ROBUST APPROACHES AND OPTIMIZATION FOR 3D DATA

A Dissertation
Presented to
The Academic Faculty

By

Rahul Sawhney

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Robotics

School of Interactive Computing
Georgia Institute of Technology

May 2018

ROBUST APPROACHES AND OPTIMIZATION FOR 3D DATA

Approved by:

Dr. Isbell, Charles L
School of Interactive Computing
Georgia Institute of Technology

Dr. Boots, Byron
School of Interactive Computing
Georgia Institute of Technology

Dr. Vela, Patricio A
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Christensen, Henrik I
Department of Computer Science and Engineering
University of California San Diego

Dr. Li, Fuxin
School of Electrical Engineering and Computer Science
Oregon State University

Date Approved: November 29, 2017

Do not walk in front of me... I may not follow
Do not walk behind me... I may not lead
Walk beside me... just be my friend
Walk away from me else... just leave me alone

Adapted, Unknown

Dedicated to my family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The thesis of this work is that — *By leveraging 3D geometry at macro scales, it is possible to perform purely geometric analysis of real world 3D data that is robust in the face of noise, viewpoint changes, occlusions and partially overlapping content.*

In this work, we introduce a robust representation framework capable of effectively harnessing macro scale 3D geometry in real world scenes; present a robust loss for improved estimation; derive a novel optimization technique for related class of nonconvex, nonsmooth losses and resulting objectives; and demonstrate the efficacy of proposed robust representations and robust optimization in demanding settings.

Sensor data from the physical world is usually unpredictable and always imperfect. The same can be said about potential application settings and possible scenarios arising in reality. Considering the challenging, often noise-ridden, nature of 3D modality itself as well — robustness becomes a practical necessity for high performing 3D-centric methods.

We present robust, purely geometric representations for fundamental association and analysis problems involving multiple views and scenes. The representations utilize surface patches / segments as the underlying data unit, and leverage 3D geometry at macro scales. We demonstrate how this results in discriminative characterizations that are robust to high noise, local ambiguities, sharp viewpoint changes, occlusions, partially overlapping content and related challenges.

We discuss a novel approach to find localized geometric associations between two vastly varying views of a scene, through semi-dense patch correspondences, and align them. We present means to evaluate structural content similarity between two scenes, and to ascertain their potential association. And we show how this can be utilized to obtain geometrically diverse data frame retrievals, and resultant rich, atemporal reconstructions.

The presented solutions are applicable over both depth images and point cloud data. They are to be able to perform in settings that are significantly less restrictive than ones under which existing methods operate. In our experiments, the approaches outperformed pure 3D methods in literature. Under high variability, the approaches also compared well with solutions based on RGB and RGB-D.

We then look at more fundamental methods to address robustness in an intrinsic sense. We introduce a robust loss function that is generally applicable to estimation and learning problems. The loss, which is nonconvex as well as nonsmooth, is shown to have a desirable combination of theoretical properties well suited for estimation (or fitting) and outlier suppression (or rejection). In conjunction, we also present a methodology for effective optimization of a broad class of nonsmooth, nonconvex objectives — some of which would prove problematic for popular methods in literature. Promising results were obtained from our empirical analysis on 3D data.

Finally, we discuss a nonparametric approach for robust mode seeking. It is based on mean shift, but does not assume homoscedastic or isotropic bandwidths. It is useful for finding modes and clustering in irregular data spaces.

CHAPTER 1

INTRODUCTION

Real world data always has noise — an external, unknown, variability which cannot be predicted. For a procedure to be utile in the physical world, it needs to be able to accommodate, tackle this perplexity — it needs to be *robust*.

Robustness is particularly necessary for the considerably challenging 3D modality. In general, the modality has high local ambiguity and may not be lavish with information on the whole (in contrast to appearance). 3D sensing data is prone to a number of imperfections as well, such as holes and spikes. This is especially true for data acquired from commodity range / depth sensing hardware which tends to be quite noisy, with several artifacts. Typical 3D data from indoor or structural environments tends to be locally smooth and isomorphic in nature, which makes the analysis significantly more difficult. Changes in viewpoint, occlusions and partially overlapping content exacerbate the problem much further, when the tasks involve multiple views and scenes.

This work discusses methods for analysis over 3D data that are robust in the face of real world challenges such as high variability and noise. The presented methods do away with certain prevailing assumptions, simplifications or limitations, that hamper performance and hinder broader or better applicability in the real world.

We consider an overarching thesis for purely geometric 3D analysis, and based on it, propose robust solutions for fundamental 3D association problems. We also discuss more abstracted, and general, methods pertaining to optimization and estimation for robust data analysis.

Our thesis is that — *By leveraging 3D geometry at macro scales, it is possible to perform purely geometric analysis of real world 3D data that is robust in the face of noise, viewpoint changes, occlusions and partially overlapping content.*

We propound the importance of *macro level geometry* for robust processing and analysis of real world 3D data. By *macro* here, we indicate geometry of a more comprehensive scope, and defined through a more holistic context — descriptions of arbitrary span over surfaces, primitives and structures, and encompassing spatial relationships between them.

The essential significance of macro level geometry is manifest in the way we perceive — be it while navigating environments, or for getting acquainted with a setting. We regularly localize, gauge positions relative to the surfaces and structures in the neighborhood — often situating ourselves, and other objects, with respect to the surrounding layout. And we

often make stronger inferences about qualitative attributes through physical form, structural characteristics and spatial configuration — like determining whether some furniture is suited for seating based on its form, distinguishing a sofa from a couch based on its structural characteristics, and discerning rooms with identical furniture or similar layout based on the specifics of the arrangement pattern.

Interestingly, quite often we can, and do, comport just fine when appearance cues are inadequate, and even when they are ambiguous or absent — macro level geometry is definitely an enabler. The invaluable metric information readily furnished by spatial geometry can empower navigation or reconstruction tasks, while recognition or inference can benefit significantly from appropriate descriptions of shape and structure.

Whilst the literature is replete with approaches dedicated to capturing 3D geometric information [1], the outstanding ones in literature have mainly been reliant on (discriminative) appearance information. This is especially the case for association tasks over multiple views and scenes. Relatively few association methodologies work well on noisy, imperfect 3D point clouds or depth images from the real world. Most of them are limited to quite specific use cases, are significantly restrictive, or critically rely on additional pieces of information to obtain strong direct or indirect association priors (for instance, [11, 12, 13, 14, 15]). There is a clear need for representations and descriptions capable of better harnessing geometry from real world 3D data, while being robust to its numerous challenges.

We consider the use of surface patches as the underlying data representation — a near complete, compact representation that affords inherent robustness to point level noise. By exploiting macro level geometry, we show how highly discriminative descriptions can be derived for / from them, without fundamentally relying on appearance. It is evinced how the developed descriptions achieve invariance to sharp changes in viewpoint, and a high degree of robustness to noises, occlusions, local ambiguities and partially overlapping content.

3D surface patches (or segments, to refer to any set of 3D contiguous surface patches) allow us to effectively represent most kinds of scenes, shapes and structures. And the ability to

[1] Most current descriptions are derived directly from / about 3D points, and their possible aggregation thereof. Although they generally afford good localization and are parsimonious, they are encumbered under viewpoint changes, point noises and local ambiguities — are not robust / stable, and often inapplicable, [1, 2]. In case of features based on interest points, poor keypoint repeatability poses another problem. Besides, descriptions based on points are not inherently suitable (or efficient) for capturing non-local geometry in face of various discontinuities and undulant curvature, particularly in everyday scenes with a multitude of interfacing bodies and surfaces. Coarse approximate representations based on planes, normal distributions, spectral analysis, fourier transforms and hough transforms have been utilized as well, in [3, 4, 5, 6, 7, 8] for instance. While decidedly more stable, they miss out on important details (like curvature and spatial layout), and do not localize well — their utility lies in coarse alignment when good initializations are available.

More recently, 3D voxel grids carrying occupancy information have been successfully employed in model-centric analysis of volumetric objects and structures, such as in [9, 10]. These are useful in reasoning about pre-localized information (known sensor poses or single views), through exhaustive description over some volume. Still, discretized occupancy cells with an anchored, global coordinate frame lack orientation - are not the most appropriate representation of projectively acquired data.

describe them discriminatively, invariantly and robustly opens up promising possibilities for association, recognition and related tasks over 3D data.

Scenarios and settings where spatio-temporal contiguity is weak, or altogether absent, could benefit from the framework's high robustness to multi-view challenges. For instance, settings involving freely / actively moving sensors, or multiple uncoordinated ones, or scenarios which involve sparse or uneven data acquisition, possibly over a network. Or search and retrieval tasks in absence of useful priors — say, from an unordered database or assorted repositories. Such scenarios may prove difficult, even unviable, for alternative 3D geometric approaches.

A purely 3D framework with good performance in the real world broadens applicability as well. Besides 3D only settings, it could be be leveraged in situations when appearance cues are inadequate or ambiguous — texture scant environs or under under weak lighting conditions, for instance. This is often the case in indoor, industrial or construction environs.

While a well designed representation can directly tackle the real world challenges of a data type and the problem domain, robustness can also be sought at a more fundamental level — abstracted from all but the core characteristics and statistics of data.

We consider robustness in this complimentary and intrinsic sense as well, through approaches for robust estimation and optimization. We delve a bit into mathematical properties that render robustness to estimator functions, on the characterization of outliers, and the significance of both nonconvexity and nonsmoothness in this regard. We thereby introduce a robust loss function that seems well suited for outlier suppression and exact model fitting. These traits are often essential for robust processing of real world 3D data.

A closely related robust optimization methodology, to solve M - estimation and structured estimation type objectives, is discussed as well. These objectives figure frequently in learning and estimation problems in perception, including ones over 3D data — such as inverse problems pertaining to motion estimation and reconstruction.

The other robust method for direct data space analysis discussed in this thesis, pertains to reliable detection of modes or bumps in the data. At times, depending on the problem at hand and the nature of feature space, an explicit characterization of the data is either unviable or unnecessarily cumbersome — denoising normals of arbitrary 3D surfaces, for instance. A simple to adapt approach, that can be applied to irregular, complex data spaces (possibly as a preprocessor) would prove useful. For such purposes, we discuss an anisotropic, data driven methodology, based on mean shift, for nonparametric mode seeking and feature space grouping / smoothing.

Thesis Overview and Chapter Outlines

1.1 Robust Anisotropic Mode Seeking

The aforementioned method for robust mode seeking and feature space smoothing / clustering is discussed first, in *Chapter* 2. It is based on mean shift — a non-parametric, derivative free, mode finding technique that has easy, general applicability. As mentioned earlier, the technique is especially handy in data spaces that are irregular, are difficult to characterize and model. It is thus useful in low-level, possibly preprocessing, tasks in perception — such as clustering or smoothing of features and noisy sensor data.

Mean shift methodologies in literature have largely been isotropic, as well as homoscedastic. We discuss how naturally guided agglomeration can address these limitations — resulting in robust, fully anisotropic and locally adaptive mode seeking / clustering. Additionally, conventional mean shift requires careful selection of the bandwidth parameter on a per instance basis. The presented approach, due to its adaptive design, also alleviates this issue - with a default form performing generally well. Analysis for convergence is covered as well.

Chapter 2 first introduces standard Mean Shift as a fixed point iteration over the kernel density estimate of data. It then discusses some of the existing work in literature in some detail, while motivating the need of a locally adaptive, anisotropic approach. The methodology is then covered in *Section* 2.2, where we derive update equations, and discuss how to agglomerate, update bandwidths and post-process. Qualitative and quantitative experiments are then covered *Section* 2.3, and we conclude with *Section* 2.4.

1.2 Robust Geometric Association with Surface Patches

We then discuss the problem of making localized geometric associations between two 3D point clouds or depth images in *Chapter* 3. This is useful for ascertaining geometric correspondences between the point sets. And when they pertain to the same scene, to calculate motion between them and align them. These problems are fundamental to several tasks in navigation, tracking, mapping and reconstruction. They are also useful in tasks involving detection and semantic content association.

We present a solution based on making potentially dense 3D surface patch correspondences between given point sets. Ascertaining surface patch correspondences in a generalized fashion, has hitherto not been addressed to our knowledge.

We show how 3D geometrical properties leveraged at macro scales, can result in a high degree of association robustness — to noise, and to issues related to viewpoint changes,

such as wide baselines, heavy rotations, significant occlusions and partial overlaps. The presented approach is based on representing patches of interest as sequences of invariant geometrical properties, by employing uniquely consistent partial orderings. These sequences are then matched through an optimal sequence alignment metric based on the Restricted Damerau-Levenshtein distance. The approach can robustly handle steep viewpoint changes while not relying on any priors on pose, motion or structure, or making any assumptions on them. In our experiments, it outperformed purely geometric baselines. Under larger viewpoint changes, it performed better than approaches based on RGB-D inputs as well.

We start *Chapter* 3 by motivating the need of a purely geometric association approach that is robust to challenges of real world 3D data, and performs competently in practice. We also explain why a framework based on surface patches can prove useful in achieving this objective. After surveying existing work in *Section* 3.1, our approach is presented in *Section* 3.2. *Section* 3.2.1 shows how patches can be expressed through a set of robust, viewpoint invariant features that capture 3D geometry at macro scales, and *Sections* 3.2.2 through 3.2.4 show how these sets can be utilized to ascertain surface patch correspondences. Surface patch segmentation and hierarchy generation is covered in *Section* 3.3. We then empirically evaluate the approach in *Section* 3.4. We not only compare it with popular 3D geometric methods, but also with state-of-the-art methods in RGB-D as well. Finally, we conclude with *Section* 3.5.

1.3 Robust Geometric Scene Association and Retrieval

We then show how to evaluate similarity between 3D point sets based on their geometric and structural content, in *Chapter* 4. Quantifying geometric similarity between views of scenes, and ascertaining whether two given views pertain to the same physical scene or not, are both fundamentally important problems. They are intrinsic to several navigation, reconstruction and recognition tasks.

We address the problem in a minimally restrictive setting — as one of geometric retrieval from an unannotated, spatio-temporally unordered database. Again, this is made possible by leveraging macro scale geometry. The approach involves expressing discriminative macro scale information in a learnt viewpoint-invariant feature space. These are then encoded in a frame-level signature that can be utilized to measure geometric content similarity.

The approach generalizes well - it does not require dataset specific training, and scales up nicely. Experiments indicated it to be robust to sharp viewpoint differences and related challenges. We also show how the methodology can be employed to affect geometric diversity — to select a set of data frames which are structurally similar yet diverse amongst themselves. We show this results in better workspace coverage and richer reconstructions.

In our experiments, the presented approach outperformed ones based on depth data. It also performed better than ones operating on RGB / RGB-D data and employing CNNs (Convolution Neural Nets).

We start *Chapter* 4 by first establishing why ascertaining 3D geometric similarity and association in a minimally restrictive setting is important. We then distinguish the problem addressed by our approach from ones in existing literature, and continue to do so in *Section* 4.1. After formalizing the problem in *Section* 4.2, we look at how a general geometric feature space can be learnt in *Section* 4.3. This utilizes feature sets that are based on *Section* 3.2.1, but are more exhaustive and have increased redundancy. *Section* 4.4 shows how features from this space can be encoded to generate signatures that can characterize scene-views. (*Sections* 4.5 and 4.6) then outline how a geometrically diverse set of retrievals can be obtained, while *Section* 4.7 shows how to ascertain physical association with some of them. Further details are covered in *Section* 4.8. *Section* 4.9 then elaborates on empirical evaluations of our approach. This includes extensive comparisons with state-of-the-art methods (mostly based on appearance). Finally, we discuss our conclusions in *Section* 4.10.

1.4 A Nonsmooth Nonconvex Loss and related Robust Optimization

Finally in *Chapter* 5, we approach robustness at a fundamental level, through robust loss functions and optimization. We introduce a robust loss, with a combination of properties well suited for 3D estimation and fitting tasks. In conjunction, we also present a methodology for optimization of nonsmooth, nonconvex objectives. The presented scheme directly addresses an important general class of losses (to which the proposed loss also belongs), and related M-estimation and structured objectives. The methodology also supports block wise optimization for increased scalability and efficiency. A nonlinear least absolute deviations solver was developed as part of the proposed framework. The solver utilizes efficient and stable proximal operations. It is useful by itself as it can address general nonlinear least absolute deviation based problems. Besides being independently useful (and much more generally applicable), the contributions in this chapter are complimentary to the front-end approaches such as ones presented in *Chapters* 3 and 4.

Chapter 5 begins by discussing the significance of nonconvex, nonsmooth formulations and why they are difficult to optimize. *Section* 5.1 then provides background on loss functions and related influence functions. The introduced loss and and its properties are then discussed in *Section* 5.2 and contrasted with a number of losses prevalent in perception literature. We then provide some background in variational factorization of loss functions in *Section* 5.3, which is utilized in our optimization methodology. Various objectives of interest are subsequently discussed in *Section* 5.4. *Sections* 5.5 through 5.7 then present specifics of the proposed optimization methodology. They detail the optimization of each of the objective

forms discussed in *Section* 5.4. Our empirical evaluations on 3D data are then presented in *Section* 5.8, and we finally conclude with *Section* 5.9.

1.5 Concluding Comments

We conclude the thesis with a summarization in *Chapter* 6. It overviews major results, and discusses some open problems and future directions.

CHAPTER 2

ROBUST ANISOTROPIC MODE SEEKING

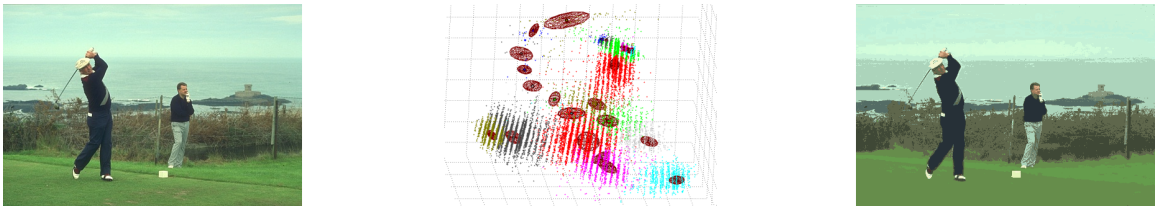
We first discuss a generally applicable method for robust analysis, for mode seeking and clustering. It is useful in low dimensional feature spaces that are irregular, are difficult to characterize and model. The methodology is based on mean shift, and allows for fully anisotropic mode seeking and clustering through unsupervised, local bandwidth selection. The bandwidth matrices evolve naturally, adapting locally through agglomeration, and in turn guiding further agglomeration. This results in increased saliency and robustness, while alleviating instance specific initialization sensitivity.

Mean shift is a powerful nonparametric technique for robust mode seeking and unsupervised pattern clustering. References [16, 17] established it's utility in low-level perception tasks such as feature clustering, filtering and in tracking. It has been in popular use since, as a very useful tool for pattern clustering of sensor data ([18, 19] for example). It has also found niche as a preprocessor (a priori segmentation, smoothing) before higher level image & video analysis tasks such as scene parsing, object recognition, detection ([20, 21, 22]). Image segmentation approaches such as Markov Random Fields, Spectral clustering, Hierarchical clustering use it as an a priori segmenter with improved results ([20, 23, 24, 25, 26]).

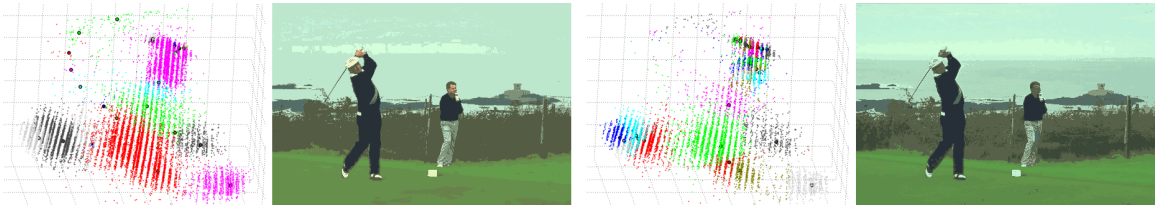
Mean Shift methodologies though, employ some assumptions and have some limitations, which may not be desirable. Its popular standard form, [16], utilizes fixed, scalar bandwidth assuming homoscedasticity and isotropicity. Being homoscedastic, it also requires proper bandwidth choice on a per instance basis. The adaptive Mean Shift variants, [27, 28], ascertain variable bandwidths, but they still assume isotropicity. They also make use of heuristics which are not flexible, and lack clustering control. Offline bandwidth selection methods for Mean Shift ([29, 30, 31]), typically estimate a single, global bandwidth, and/or are data specific/non-automatic. As indicated in *Figure* 2.1 - isotropic/scalar bandwidths tend to smooth anisotropic patterns and affect partition boundaries, while global/homoscedastic bandwidths are inappropriate when clusters (or modes) at different scales need to be identified.

We present a mean shift methodology which is anisotropic and locally adaptive. It is able to leverage guided agglomeration for unsupervised bandwidth selection (*Figure* 2.1). This results in robust mode detection, with increased partition saliency. Also as a consequence, a low valued parameter set performs nicely over a wider range of data instances (*Section* 2.2.1). We also present a useful result in [32] - a convergence proof when full bandwidths vary

Figure 2.1: **Exemplar result with comparisons :** Indicative, illustrative result of our approach, AAAMS (a), is shown along with conventional Mean Shift results (b), at comparable clustering levels. As is indicated by the plots and segment images, AAAMS effectively adapts to local scale and preserves anisotropic details, affecting more salient partitions.



(a) 3D Clustering result (23 clusters) over image data (left, *L*a*b** space) by the proposed approach. *1-sigma* final trajectory bandwidths have been overlaid over the converged modes. The segment image is shown on right.



(b) Comparitive results with standard MS (left) and variable-bandwidth isotropic MS ([28], right), at similar clustering levels, 25 & 27 respectively, are shown. Final mode locations have been indicated over the cluster plots. MS with correctly chosen bandwidth detected more coherent modes than [28], but looses partition saliency (bushes, water, sky in background). [28] better adapts to scales but oversegments at places, and smooths over others (face). Both smoothed over details, failed to detect some modes at lower scales (trouser edges, maroon on shirt & shoes). In general, conventional MS had a typical tendency to over-segment heavily or compromise partition boundaries.

between Mean Shift iterations, as is the case here.

Clusters arise on the fly in the proposed approach, as a consequence of agglomeration of extant clusters. *Local bandwidths* (*Sections* 2.1 and 2.2) which evolve anisotropically every iteration, are associated with each cluster; by design, all members of a cluster converge to the same local mode. By evolving as function of a cluster's aggregated trajectory points, these bandwidths are able to adapt to the underlying mode structure (shape, scale, orientation) - and in turn, guide future cluster trajectory and agglomeration. We refer to our approach as online because it's an on the fly unsupervised procedure; with simple bookkeeping doing away with re-calculations.

2.1 Motivation and Background

We utilize the exposition style of [33]. Let $\{x_i\}_{i=1}^n \subset R^d$, be a set of *d*-dimensional data points with their sample point kernel density estimate (KDE) being $p(x) = \sum_{i=1}^n p(x_i)p(x|x_i) = \sum_{i=1}^n p_i \frac{1}{c_i} K(\|x - x_i\|_{\Sigma_i})$. Stationary points of the KDE can be estimated by evaluating the density gradient and setting it to zero. This gives rise to the Mean-Shift fixed point iteration :

$$x^{\tau+1} = f(x^\tau) \tag{2.1a}$$

$$f(x^\tau) = \left(\sum_{i=1}^n p_i \frac{1}{c_i} K^{'}(\|x^\tau - x_i\|_{\Sigma_i}) \Sigma_i^{-1} \right)^{-1} \times \left(\sum_{i=1}^n p_i \frac{1}{c_i} K^{'}(\|x^\tau - x_i\|_{\Sigma_i}) \Sigma_i^{-1} x_i \right) \tag{2.1b}$$

$K(t)$, $t \geq 0$, is a *d*-variate kernel with compact support satisfying some regularity constraints, mild in practice ([16, 33] for details). $\|x - x_i\|_{\Sigma_i} \equiv ((x - x_i)^T \Sigma_i^{-1} (x - x_i))^{1/2}$, is the Mahanalobis metric. The point prior $p_i \equiv p(x_i)$ is usually taken as $1/n$. c_i is a normalizing constant depending only on the covariance matrix, Σ_i (kernel bandwidth), associated with each data point. The bandwidth, Σ_i, is roughly an inverse measure of local curvature around x_i. It linearly captures the scale and correlations of the underlying data. τ indicates the iteration count. In practice, since $K(t)$ is taken with truncated support, the summations are only over n' neighbors of x^τ, with $n' \ll n$. The vector $m(x^\tau) = f(x^\tau) - x^\tau$, is referred to as the Mean Shift. It's a bandwidth scaled version of $\nabla p(x)$, is free from a step size parameter, is large in regions with low $p(x)$ and small near the modes. Starting at a data point, $x_i^{\tau=0} \equiv x_i$, the fixed point update is run multiple times till convergence. The resulting points, $x_i^{\tau \geq 1}$ is referred to as the *trajectory* of x_i, tracing a path to the local mode. The technique thus, is able to locate modes and partition feature space, without a priori knowledge of partition count or structure.

The above hinges on selecting reasonable bandwidth matrices Σ_i. Good bandwidths capture the underlying local distribution effectively. In our approach, data points (pertaining to a cluster) converging to a common local mode share a common bandwidth - one which reflects this mode's structure, and to an extent, its basin of attraction ([16]). We refer to it as the local bandwidth ([31] utilizes local bandwidths in a related sense).

In online unsupervised usage, almost all Mean Shift variants for clustering, for example [16, 26, 34, 35], work under the restrictive assumptions of homoscedasticity and isotropicity ($\Sigma_i = \sigma^2 I$, standard fixed bandwidth Mean Shift). The scale parameter σ has to be set carefully based on the dataset instance. [36] utilizes set covering based iterative agglomeration for improved efficiency. Coverage is ensured through overlaps of small fixed homoscedastic bandwidths. Some applications only assume isotropicity ($\Sigma_i = \sigma_i^2 I$, adaptive

/ variable-bandwidth Mean Shift). σ_i is estimated using a variation of the following two heuristics ([27, 28]) - 1) k^{th} nearest neighbor, x_i^k, distance heuristic $\rightarrow \sigma_i \propto \|x_i - x_i^k\|$, or 2) Abramson's heuristic $\rightarrow \sigma_i \propto \sigma_o(\pi(x_i))^{-1/2}$, where $\pi(x)$ is the *pilot* density estimate obtained by first running mean shift with analysis bandwidth, σ_o. They have found more use in smoothing type applications as reported in [37, 38]. Variants have also been used in tracking scenarios, where the bandwidths are adapted in a task specific fashion (see [39, 40], for example). [41, 42] adapt isotropic bandwidths to object scales, to unimodally track, search for them. The topological, blurring, evolving variants for clustering (like [35, 26, 34, 43, 44]) use isotropic bandwidths. They are primarily aimed at increased efficiency, with results on par with standard mean shift. [45] presents improvements over the somewhat related Mediod Shift. They propose usage of their algorithm as initialization for Mean Shift, for increased efficiency.

In offline settings, [31] presents a supervised methodology. Training data is processed with analysis bandwidths to select local bandwidths based on neighboring partition stability. The estimated bandwidths are then used to partition similarly distributed test image data. Only recently were automatic full bandwidth selectors for density gradient estimation proposed in [30, 29], for offline settings. These focus on obtaining good data density gradients (as opposed to clustering) and optimize based on the mean square integrated error (MISE). A single global bandwidth is estimated for the given data, and as the authors themselves note, the involved computations are not straighforward.

A very useful variant is Joint Domain Mean Shift, [16], which is used to create partitions jointly respecting the dataset's multiple feature domains which are mutually independent; For example, $\langle color, space \rangle$ in color based segmentation & smoothing, and $\langle color, flow \rangle$ in motion segmentation. When $x = [{x^r}^T \, {x^s}^T]$ with $(x^r \perp x^s) \, | x$, and utilizing two separate kernels, K_r, K_s, we'll have $p(x) = \sum_{i=1}^{n} p(x_i) p(x^r | x_i) p(x^s | x_i)$. Equation 2.1b analogue would then be $f(x^\tau) = \left(\sum_{i=1}^{n} p_i \frac{1}{c_i} J(\|x^{\tau,r} - {x_i}^r\|_{\Sigma_i^r}, \|x^{\tau,s} - {x_i}^s\|_{\Sigma_i^s}) \Sigma_i^{-1} \right)^{-1} \times \left(\sum_{i=1}^{n} p_i \frac{1}{c_i} J(\|x^{\tau,r} - {x_i}^r\|_{\Sigma_i^r}, \|x^{\tau,s} - {x_i}^s\|_{\Sigma_i^s}) \Sigma_i^{-1} x_i \right)$, where $c_i^{'}$ is the normalization constant, $J(t_1, t_2) \equiv K_r'(t_1) K_s(t_2) = K_r(t_1) K_s'(t_1), \forall t_1, t_2 \geq 0$, and $\Sigma_i = \begin{bmatrix} \Sigma_i^r & 0 \\ 0 & \Sigma_i^s \end{bmatrix}$. Typically, but not necessarily, x^s may lie on a spatial manifold - imposing structure to data which is utilized. Instances in literature use fixed global scale parameters σ^r and σ^s, which have the aforementioned limitations. As noted in [25] on color segmentation, σ^r and σ^s need to be selected carefully. Good choices are not always possible, with segments being too coarse or too fine at times (*Figs. 2.4*). Reference [46] utlizes an anisotropic Σ_i^s for visual data segmentations. Every data point's associated bandwidth, Σ_i, is modulated multiple times in each iteration, until convergence is achieved. Modulation heuristics have been provided, to be deployed as per task. The spatial bandwidth Σ_i^s is parameterized as function of eigenvectors of neighborhood data covariance. Σ_i^r is taken to be an isotropic scalar dependent on Σ_i^s.

Algorithm 1 : AAAMS - Anisotropic Agglomerative Adaptive Mean Shift

---

$Function\ :\ AAAMS\left\langle \{x_i\}_{i=1}^{n} \right\rangle\ with\ x_i \in R^d$

$Returns\ :\ \langle U^*,\ C^*,\ \{\mu_u\}_{u \in U^*},\ \{\Sigma_u^*\}_{u \in U^*} \rangle$

$\#\#\ Convergence\,Criteria\ \rightarrow\ \|m_u\| \leq \delta$

$U^o = \{1, \ldots n\}\ ;\ C_u = \{x_u\}\ ,\ x_u^o = x_u,\ \Sigma_u = \sigma_{base}^2 I_d,\ \forall u \in U^0$

$\tau = 0\ ;\ \lambda = 5\ ;\ \delta = Convergence\ epsilon$

$m_u = Large\ \in\ R^d,\ T_u = \phi\ ,\ \forall u \in U^0$

$While\ \exists u\ \in U^\tau s.t.\ \|m_u\| > \delta$

$ForEach\ \ u\ \in\ U^\tau\ s.t.\ \|m_u\| > \delta$

$$u^{\tau+1} = \begin{cases} Eq.\,2.4 & ESS(u)\ <\ \lambda \\ Eq.\,2.2b & ESS(g)\ \geq \lambda,\ \forall g \in G \\ Eq.\,2.3 & otherwise \end{cases}$$

$m_u = u^{\tau+1} - u^\tau\ ;\ T_u = T_u \cup u^{\tau+1}$

$Get\,Ne_x(u^{\tau+1})$

$ForEach\ \ y\ \in\ Ne_x(u^{\tau+1})\ or\ till\ C_u \neq \emptyset$

$If\ \Pi(y)\ = u\ or\ C_{\Pi(y)} = \phi\ \ Then\ \ Continue$

$If\ \ \left\|u^{\tau+1} - y\right\| > \epsilon\ \ Then\ \ Continue$

$If\ !MergeCheck\left\langle u^{\tau+1}, y, m_u, y^{\tau=1} - y, u, \Pi(y) \right\rangle$

$Then\,Continue$

$If\ \rho_u > \rho_{\Pi(y)}$

$Then$

$C_u = C_u \cup C_{\Pi(y)}\ ;\ C_{\Pi(y)} = \emptyset$

$T_u = T_u \cup T_{\Pi(y)}\ ;\ T_{\Pi(y)} = \emptyset$

$Else$

$C_{\Pi(y)} = C_u \cup C_{\Pi(y)}\ ;\ C_u = \emptyset$

$T_{\Pi(y)} = T_u \cup T_{\Pi(y)}\ ;\ T_u = \emptyset$

$EndForEach$

$If\ C_u = \phi\ \ Then\ \ Continue$

$$\Sigma_u = \begin{cases} Eq.\,2.6 & ESS(u)\ \geq\ \lambda \\ \Sigma_u & otherwise \end{cases}$$

$\#\#\ Optionally\ Perturb\left\langle u^{\tau+1},\ m_u \right\rangle\ \ if\ \ \|m_u\| \leq \delta$

$EndForEach$

$U^{\tau+1} = \{u \mid u\ \in\ U^\tau,\ C_u \neq \emptyset\}$

$\tau = \tau + 1$

$EndWhile$

$U^* = U^\tau\ ;\ \ C^* = C_u\ ,\ \forall u \in U^\tau\ ;\ \ \ \Sigma_u^* = \Sigma_u\ ,\ \forall u \in U^\tau$

$EndFunction$

For feature spaces that can be decomposed into independent subspaces, the above can be extended to multiple domains. The update equations would then utlize multiple kernels. Basically, for each domain, a $\langle \sigma_{base},\ \epsilon \rangle$ *pair needs to be set.*

*For example, for joint domain Mean Shift (*Section *2.1), we'll have* $\langle \sigma_{base}^r,\ \epsilon^r \rangle$ & $\langle \sigma_{base}^s,\ \epsilon^s \rangle$ *for the two domains. We'll have then* $\Sigma_{base} = \begin{bmatrix} {\sigma_{base}^r}^2 I_r & 0 \\ 0 & {\sigma_{base}^s}^2 I_s \end{bmatrix}$ & $\Sigma_u = \begin{bmatrix} \Sigma_u^r & 0 \\ 0 & \Sigma_u^s \end{bmatrix}$. Σ_u^r & Σ_u^s *would be evaluated from Equation 2.6. Equation 2.2b analogue would be* $f(u^\tau) = \left(\sum_{\forall g \in G} \frac{1}{c_g} \Sigma_g^{-1} \sum_{\forall i | x_{i,g} \in Ne_x(u^\tau)} J(\|u^{\tau,r} - {x_i}^r\|_{\Sigma_u^r},\ \|u^{\tau,s} - {x_i}^s\|_{\Sigma_u^s}) \right)^{-1} \times$ $\left(\sum_{\forall g \in G} \frac{1}{c_g} \Sigma_g^{-1} \sum_{\forall i | x_{i,g} \in Ne_x(u^\tau)} J(\|u^{\tau,r} - {x_i}^r\|_{\Sigma_u^r},\ \|u^{\tau,s} - {x_i}^s\|_{\Sigma_u^s}) x_i \right)$; *likewise for others.*

2.2 Methodology

A data point, x_i, is alternatively represented as $x_{i,u}$ - the first index value being its unique identifier as before and the second index indicating its current, exclusive membership to a cluster, $u \in \{1, \dots n\}$ (the second index is left out when the membership is apparent or inconsequential). A cluster u's constituent data points is denoted by the set , $C_u = \{x_{i,u} \mid \exists\, i \in \{1, \dots n\}\}$. By algorithm design, clusters are merged only when they are tending towards the same mode - thus all member points of a cluster, u, will eventually converge to a common local mode, say μ_u. They hence, are also taken to share a common local bandwidth, Σ_u. This bandwidth develops every iteration when the cluster u's trajectory points set, T_u , gets additional elements. The set of clusters surviving at iteration, τ, would be $U^\tau = \{u \mid C_u \neq \emptyset\}$. $|U^\tau|$ would indicate its cardinality. At beginning, at $\tau = 0$, each point trivially forms a separate cluster $\rightarrow U^0 = \{1, \dots n\}$, $C_u = \{x_{i=u,u}\}, \forall u \in U^0$. Given the initialization, each extant cluster $u \in U^\tau$ will always contain the initial point, $x_{u,u}$ - which we refer to as its *principle member*.

At any iteration τ, for each extant cluster u, mean shift updates happen for only the principle member, $x_{u,u}$; with the first iteration running over trivial clusters. The resulting trajectory is specified as $x_{u,u}^{\tau \geq 1}$ or simply u^τ. A cluster's trajectory might end when it gets merged or converged. In general, each data point, x_i, started out as a trivial cluster, and had or still has a trajectory - it's trajectory set being $\{x_i^{\tau=1:end}\}$. $'end'$ being the iteration at which the trajectory ended; else the current iteration. Note that the data point x_i itself is not included in this set. For any surviving cluster *u*, then, the complete set of all agglomerated trajectory points associated with it, would be $T_u = \{\cup \{x_i^{\tau=1:end}\} \mid x_i \in C_u\}$ - basically a union of all the members' trajectory sets. u^τ is indicative of the cluster u's location. At convergence, u^τ would be the location of a local mode. u's members would then be comprising of data points pertaining to that mode and its basin (*Figure* 2.1a). The data density in the immediate vicinity of u^τ's current position is indicated as $\rho(u^\tau)$, or simply, ρ_u. We use operator Π to retrieve the cluster identifier of an arbitrary data point; so $\Pi(x_{i,u}) = u$. The n' data points in u^τ's neighborhood are denoted as $Ne_x(u^\tau)$, and the clusters containing them as $G = \{\cup \Pi(y) \mid y \in Ne_x(u^\tau)\}$.[1]

The methodology for anisotropic, agglomerative, adaptive Mean Shift (AAAMS) is presented as a pseudo code in *Algorithm-*$AAAMS$. At every iteration, the following steps are run for each surviving cluster that has not converged

1) Mean shift update is computed and the cluster's location is updated. No merges happen before the

[1] Since a cluster corresponds one-to-one with its principle member, principle member's trajectory is at times referred to as cluster trajectory. Similarly, convergence of trajectory is referred to as cluster converging. τ, apart from indicating iteration, also differentiates between a trajectory point and a data point. The cluster trajectory, $u^\tau \equiv x_{u,u}^{\tau \geq 1}$, is the trajectory of data point $x_{u,u}$. The cluster u's current location refers to current position of the principle member, indicated by u^τ.

first update.

2) Nearest neighbors about the current location are ascertained - they are utilized for cluster merges, and for the mean shift update in subsequent iteration.

3) When merge criteria are met, either some clusters (owners of the neighborhood points which lie within epsilon) get merged into this cluster, or this cluster gets merged into one of them.

4) If the incumbent cluster survived after the merge, its bandwidth is updated.

5) Optionally, if the cluster has converged, its location could be perturbed a bit. It is, then, not taken out of consideration in subsequent iteration.

2.2.1 Update Equations

Taking $p_i = 1/n$ and limiting summations to the neighboring points, $Ne_x(u^\tau)$, the fixed point iteration, Equations 2.1a and 2.1b, over a cluster u (rather $x_{u,u}$) can be reformulated/reorganized as a local bandwidth based decomposition :

$$u^{\tau+1} = f(u^\tau),\ where\, u^{\tau=0} \equiv x_{u,u} \tag{2.2a}$$

$$f(u^\tau) = \Bigg(\sum_{\forall g \in G} \frac{1}{c_g} \Sigma_g^{-1} \sum_{\forall i | x_{i,g} \in Ne_x(u^\tau)} K^{'}(\|u^\tau - x_i\|_{\Sigma_g}) \Bigg)^{-1} \dots \\ \dots \times \Bigg(\sum_{\forall g \in G} \frac{1}{c_g} \Sigma_g^{-1} \sum_{\forall i | x_{i,g} \in Ne_x(u^\tau)} K^{'}(\|u^\tau - x_i\|_{\Sigma_g})\, x_i \Bigg) \tag{2.2b}$$

Equation 2.2b would be exactly the same as Equation 2.1b at $\tau = 0$, when all points form trivial clusters. When local homoscedasticity in neighborhood of u^τ is assumed with the cluster's own bandwidth Σ_u taken as bandwidth estimate for neighborhood $Ne_x(u^\tau)$, Equation 2.2b simplifies [2] to :

$$f(u^\tau) = \frac{\sum_{\forall i | x_i \in Ne_x(u^\tau)} K^{'}(\|u^\tau - x_i\|_{\Sigma_u}) x_i}{\sum_{\forall i | x_i \in Ne_x(u^\tau)} K^{'}(\|u^\tau - x_i\|_{\Sigma_u})} \tag{2.3}$$

If global homoscedasticity and isotropicity is assumed, Equation 2.2b takes the form of

[2] Equation 2.3 gets us a particularly insightful interpretation. Note that $\|u^\tau - x_i\|_{\Sigma_u}$ could be thought of as a partial likelihood measure of the data point x_i belonging to the cluster u. Consider the conditional $\rightarrow\ p(x_i/u^\tau; u) = {K^{'}(\|u^\tau - x_i\|_{\Sigma_u})}/{\sum_{\dots} K^{'}(\|u^\tau - x_i\|_{\Sigma_u})}$, with the summation in denominator normalizing the distribution. The fixed point update from Equation 2.3 would then come out to be $u^{\tau+1} = \sum_{\dots} p(x_i/u^\tau; u) x_i$. So the updated cluster trajectory $u^{\tau+1}$ is just the neighborhood data expectation, conditioned only under the cluster's own distribution. In effect, this serves to guide/update a cluster's trajectory based only on the properties (bandwidth) it has itself ascertained (till τ).

standard mean shift update, where the bandwidth is specified through a fixed scalar σ_{base} :

$$f(u^\tau) = \frac{\sum_{\forall i | x_i \in Ne_x(u^\tau)} K^{'}(\|{(u^\tau - x_i)^T (u^\tau - x_i)}/{\sigma^2_{base}}\|)\, x_i}{\sum_{\forall i | x_i \in Ne_x(u^\tau)} K^{'}(\|{(u^\tau - x_i)^T (u^\tau - x_i)}/{\sigma^2_{base}}\|)} \tag{2.4}$$

Each trivial cluster utilizes fixed base bandwidth to begin with, employing Equation 2.4 for mean shift updates. Benign clusters form and start moving up on some modes. As soon as a cluster accumulates enough trajectory points for full bandwidth estimates (*Sec.*2.2.2) to be significant (u has moved up to denser regions by then), it switches to anisotropic updates, given by *Eqs. 2.3* & *2.2b*. A reasonable test of significance for Σ_u estimates, is to check if the kernel weighted point count or *Effective Sample Size* (ESS, [47]) is above some value, λ.

$$ESS(u) = \frac{\sum_{\forall v \in T_u} K^{'}(\|\overline{T_u} - v\|_{\Sigma_u^{estimate}})}{\sum_{\forall v \in T_u} K^{'}(\|0\|_{\Sigma_u^{estimate}})} \tag{2.5}$$

$\overline{T_u}$ indicates the mean of the trajectory set. The anisotropic update Equation 2.3 is used when the cluster has an $ESS(u) \geq \lambda$, and the more confident update Equation 2.2b is used, when $ESS(g) \geq \lambda,\ \forall g \in G$ - when all the neighboring clusters too have confident enough bandwidth estimates [3]. As a binomial rule of thumb ([47]), $\lambda = 5$ is chosen as the minimum ESS, which is analogous to choosing 5 as the minimum individual expected cell counts in a χ^2 test of independence.

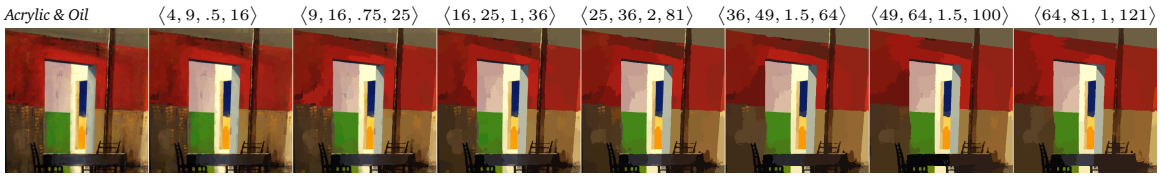
So starting with the initial base scalar, σ_{base}, the bandwidth matrices evolve by themselves. The nice part is that just a low base value suffices for reasonably dense data, with the bandwidths scaling data driven thereon and adapting to the local structure's scale, shape and orientation. σ_{base} thus becomes indicative of the minimum desired detail in the data space. This is opposed to traditional Mean Shift - where the bandwidth scalar is indicative of the scale at which the data space has to be partitioned.

2.2.2 Bandwidth Estimation

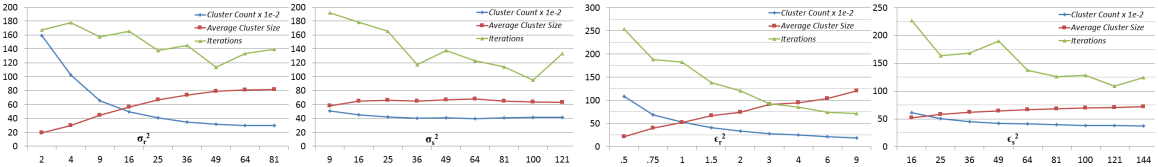
Bandwidth estimates based on a cluster's member data point locations are not reliable ([31] notes this too). A subset of point locations in isolation cannot be considered as representative of underlying distribution. The underlying local distribution is actually a localized subset of the joint non-parametric density represented by the entire dataset - it has significant contributions from neighboring structures as well. The local structure could also be asymmetric and/or without tail(s). A solution lies in considering points which arise from

[3] We note empirically for dense data, as in images, a simple cluster size sufficiency check works well. For joint domains, a cluster could switch to anisotropic updates when it has atleast $\max\left(dim(x^r),\ dim(x^s)\right)^2$ members.

Figure 2.2: **Control with parameter variation :** For joint domain AAAMS over images, we show the qualitative and quantitative effects of varying the detail and vicinity parameters, $\langle {\sigma^r_{base}}^2, {\sigma^s_{base}}^2, \epsilon_r^2, \epsilon_s^2 \rangle$. Post processing was disabled, except for enforcing cluster contiguity. As can be seen, if the need be, a good control over smoothing and segmentation levels can be exercised.



(a) Effects of varying the detail and vicinity parameters on a brush painting with smudged colors.



(b) Parameter sensitivity plots. Each of $\langle {\sigma^r_{base}}^2, {\sigma^s_{base}}^2, \epsilon_r^2, \epsilon_s^2 \rangle$ was varied while keeping others constant. Their effects on number of clusters, their average size, and iterations for convergence are plotted. Results were averaged over 33 images. As with conventional MS, color domain parameters are understandably more sensitive. $\delta = .01$ was used.

mean shift ascents over the mode the cluster is converging to - the cluster trajectory set, T_u. We use the variance of T_u with respect to the underlying density as an estimate, Σ_u. As T_u builds up each iteration, so does Σ_u.

$$\Sigma_u = \frac{\sum_{\forall v \in T_u} \rho(v) v v^\tau}{\sum_{\forall v \in T_u} \rho(v)} - \eta_u \eta_u^T + \xi I, \;\; where \, \eta_u = \frac{\sum_{\forall v \in T_u} \rho(v) v}{\sum_{\forall v \in T_u} \rho(v)} \tag{2.6}$$

$\rho(v)$ is the data density in the immediate vicinity of a point $v \in T_u$. This is evaluated using σ_{base} for consistency across clusters. $\eta_u \,\&\, \Sigma_u$ are then basically the expectation and variance of the localized distribution. In practice, a small regularizer, ξ, has to be added to the diagonals of Σ_u to prevent degenerate fitting in sparse regions, and for numerical stability.While computing anisotropic updates, eigenvalue decomposition is employed and any eigenvalues of Σ_u which fall below ξ , are clamped to it. Note that Σ_u always remains positive definite. Also note that all summations are computed on the fly.

Equation 2.6 could also be thought of as density weighted trajectory set variance. As a cluster approaches a mode, mean shift trajectory points get more concentrated and are weighted more, leading to a conservative but more localized and robust estimate – more immune to long tails. *Figures* 2.1 and 2.3 plot the bandwidths and modes at convergence, for color and point data.

(a) Simulated data in $2D$ and $3D$

(b) $L * a * b*$ space clustering

Figure 2.3: **Mode detection, clustering and final bandwidths :** Examples over color data (top row, 11 clusters) and simulated gaussian mixtures (second row) in 2D & 3D respectively. $1 - sigma$ final trajectory-set bandwidths have been overlaid at converged mode positions.

2.2.3 Cluster Merging

For any given data points, if their mean shift trajectories intersect, they will converge to a common local mode. Thus in the vicinity of a data point's trajectory (which is moving up some mode) - any data points in sufficient proximity, having their shift vectors deemed to be intersecting with this trajectory, could be clustered together. They will eventually end up converging on the same local mode. So we basically consider the data points in the vicinity of a cluster trajectory, u^{τ} - with an epsilon ϵ, delineating the vicinity. If a data point, y, in vicinity is ascertained (in *MergeCheck*) to be heading to the same mode as u^{τ}, then by transitivity - all the members of its parent cluster, $\Pi(y)$, are heading to that mode too - the clusters u and $\Pi(y)$, can then be merged. The cluster which is higher up the mode (higher density) assimilates the other cluster into itself, thus accelerating convergence to the mode. This also helps in avoiding spurious merges.

MergeCheck - This is intentionally specified as a generic function returning a true/false value. It could be implemented to suit different feature spaces and clustering criteria. The more holistic this check is, the larger the operating range of ϵ can be (assuming the distance norm holds up), without impacting clustering stability. In our experiments, we used a very lightweight generic implementation that worked well over considered data spaces - basically verifying through inner product checks that 1) relative distance between $u^{\tau+1}$ and y is decreasing and 2) Mean shift bearings [4] at $u^{\tau+1}$ and y are in the same direction. We note though that divergence measures like Bhattacharya (*Sec.* 2.2.4), kernel induced feature space metrics ([23]), information-theoretic ones like Renyi's entropy ([19]) seem viable, interesting possibilities for MergeCheck. We are yet to experiment with them.

2.2.4 Post Processing

Once data has been partitioned, a post processing step merges clusters with proximate modes, and ensures a minimum cluster size (in conventional Mean Shift, clusters are delineated

[4]The bearing at $u^{\tau+1}$ is m_u. The bearing at y, given by $y^{\tau=1} - y$, is the mean shift vector resulting from the first iteration over the trivial cluster containing y; it's stored up for consequent use.

only during the post process). Additionally for structured data, cluster contiguity could be enforced. We use graph operations. For structured data as in images, adjacency connections between clusters can be added naturally using a spatial grid structure. For unstructured data, connections between a cluster and all clusters within a reasonably large distance threshold (mode to mode distances) were added, to ensure a connected graph. Bhattacharya divergence ([48], d_B) was used as the merging criteria. It takes into account not just the variance normalized mode proximity, but also the disparity in variances themselves (*Mahanalobis* measure is its special case). $0 \leq d_B \leq 4$ was a good range, with $d_B = 1$ (somewhat analogous to $1 - sigma^2$ disparity) performing well generally [5]. The steps are shown below.

a) (For structured data only) For each cluster, use spatial adjacency to ascertain the disconnected components (highest density/mode locations for these small disconnected point sets need to be recomputed). Each disconnected component forms an additional separate cluster thereon.

b) Build the adjacency graph.

c) Merge all clusters which fall below minimum desired size, to the closest adjacent cluster until no such remain.

d) For each remaining cluster, using its constituent points, compute the density weighted variances, similar to Equation 2.6 - this is representative of the cluster's stand-alone distribution and alleviates tail influences.

e) For each pair of remaining clusters $\{a, b\}$, connected by an adjacency edge, evaluate d_B. If it falls below a certain threshold, merge the two.

$$d_B = \frac{1}{8}\left(\mu_a - \mu_b\right)^T \left(\frac{\Sigma_a + \Sigma_b}{2}\right)^{-1} \left(\mu_a - \mu_b\right) + \frac{1}{2} ln \left(\frac{det\left(\frac{\Sigma_a + \Sigma_b}{2}\right)}{\sqrt{det\left(\Sigma_a\right).det\left(\Sigma_b\right)}}\right) \tag{2.7}$$

2.3 Results

The base scalar parameter σ_{base}, in effect, regulates the minimum desired detail in the feature space, the smoothing level. The vicinity parameter, ϵ, regulates cluster merge chances and hence cluster sizes. For images, with AAAMS operating over joint domains of $\langle color,\ space \rangle$, the detail and vicinity parameters would be $\langle \sigma^r_{base},\ \sigma^s_{base} \rangle$ and $\langle \epsilon_r,\ \epsilon_s \rangle$ respectively (indicated in *Algorithm-*$AAAMS$). *Figure* 2.2, shows quantitative and qualitative effects of their variation. Although a good degree of control is possible to achieve a desired result, our experiments showed that any low valued set gave nice results over a good range of images.

Due to agglomeration, the number of clusters decrease monotonically every iteration. Only a fraction of clusters remain after the first couple of iterations; with the cluster count falling rapidly in all early iterations. The scheme thus results in a drastic reduction in net mean shift computes - as compared to the hitherto style of clustering only after convergence,

[5] For images, since color similarity alone is of consequence, d_B was evaluated only over the $L^*a^*b^*$ space

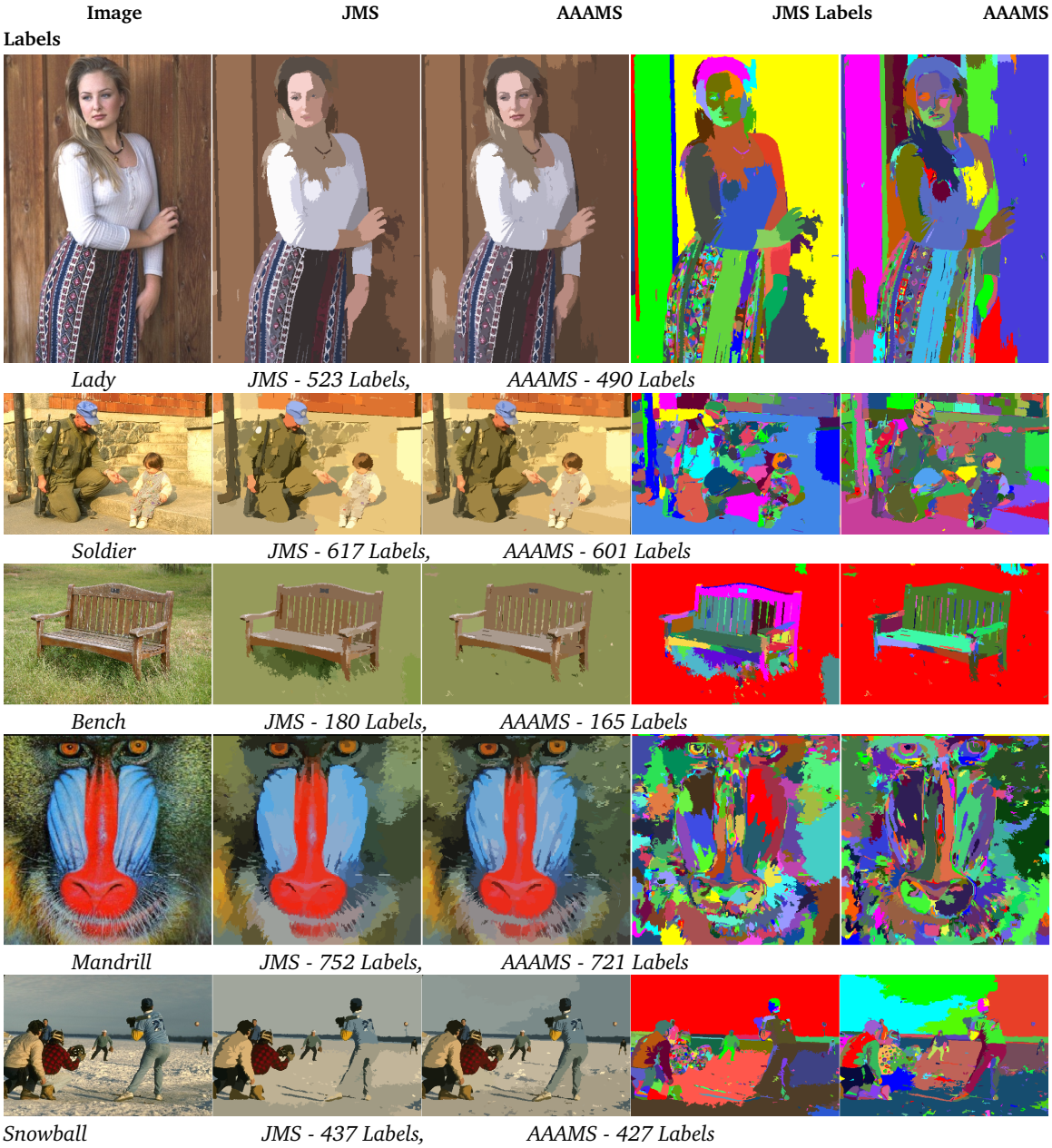


Figure 2.4: **Results and comparisons over image data :** AAAMS preserves more details and affects more perceptually salient segmentations than joint domain mean shift, at similar clustering levels. We used a single parameter set, $\langle {\sigma^r_{base}}^2, {\sigma^s_{base}}^2, \epsilon_r^2, \epsilon_s^2 \rangle = \langle 15, 16, 1, 81 \rangle$ with $d_B = 1$, to show its adaptivity on varied images. JMS segments were kept around the same, with eye on preserving detail; it still smooths over at places. Its parameter values varied significantly from image to image - $\sigma^{r\,2} \in [49,\ 81]$, $\sigma^{s\,2} \in [100,\ 289]$. Minimum cluster size was 10.

where computations happen for every data point, in each iteration. (for dense image data, typically less than 5% of the clusters remain by the 11^{th} or 12^{th} iteration). This serves to offset the additional computational workload arising from the use of full bandwidth matrices. Our straight up joint domain implementation was achieving similar timings on average

Table 2.1: **Quantitative results on BSD300 ([49]) dataset :** We used a single parameter set $\langle 20, 36, 1, 64 \rangle$ for AAAMS. For better results, d_B was set from $\{.25, .5, 1, 1.25, 1.5, 2\}$. JMS* parameters were selected per image to maintain similar segmentation levels, with an eye on preserving details, segment saliency. For perspective, we also reproduce results from [24] of unsupervised image segmentation methods. [24] selects segment levels per image. Top three values for each index are colored as *rgb*. AAAMS performs best overall - it's clearly ahead in PRI & GCE, and is a close second in BDE. Note that [24], which has the next best values, operates over *a priori* Mean Shift segmentations.

Methods / Score	***PRI***	***GCE***	***VoI***	***BDE***
$AAAMS$	.8230	.1589	2.1785	12.60
JMS^*	.7870	.1608	2.2484	13.34
Prior Art [24]				
$FullSpectralOverMS$ [24]	0.8146	0.1809	1.8545	12.21
JMS [50]	0.7958	0.1888	1.9725	14.41
$NCut$ [24] - Ref. [27]	0.7330	0.2662	2.6137	17.19
$MNCUT$ [24] - Ref. [6]	0.7632	0.2234	2.2789	13.17
$GBIS$ [24] - Ref. [9]	0.7139	0.1746	3.3949	16.67
$Saliency$ [24] - Ref. [8]	0.7758	0.1768	1.8165	16.24
$JSEG$ [24] - Ref. [7]	0.7756	0.1989	2.3217	14.40

to standard Mean Shift, which uses scalar bandwidths. Improvements in efficency based on fast nearest neighbor search such as exploiting grid structure of spatial domain, locally sensitive hashing ([28]) are applicable in our methodology too. Using Gaussian kernels, with a convergence delta, δ, set adequately to $.01$, merges would cease before 90^{th} iteration, with convergence around the 100^{th}. When just pre-partitioning is the end objective, the merging scheme thus allows us to fine tune stopping criteria. Along with the first iteration shift vectors, globally normalized local density values at each data point were stored for consequent use too. In each iteration, $\rho(u^\tau)$ was then approximated by the density value at u^τ's nearest data point. We found perturbations to be generally useful, lending to mode detection robustness and more salient partitioning. A cluster at convergence can be perturbed a fixed number of times consecutively, with progressively damped magnitudes. u then, would not be brought out of contention in the next iteration - although the immediate trajectory point resulting from the perturbation will not be included in T_u. The results presented in this paper though, are with perturbations disabled.

For image data, comparisons (*Figure* 2.4, *Table* 2.1 [6]) are shown with joint domain Mean

[6]Probabilistic Rand Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE), Boundary Displacement Error (BDE). The first three are clustering purity measures. PRI is a measure of the fraction of pairs of points whose labels are consistent with a given labeling. VoI and BDE are relative distance metrics between two given segmentations, based on average conditional entropy and boundary pixel difference, respectively. GCE measures the extent to which one labeling can be viewed as a refinement of the other. Higher is better for PRI while lower is better for the other three. For BSD300, the values indicate how well a segmentation corresponds to ones by human subjects. We noticed that coarser segmentatios tended to give better values. This, we suppose,

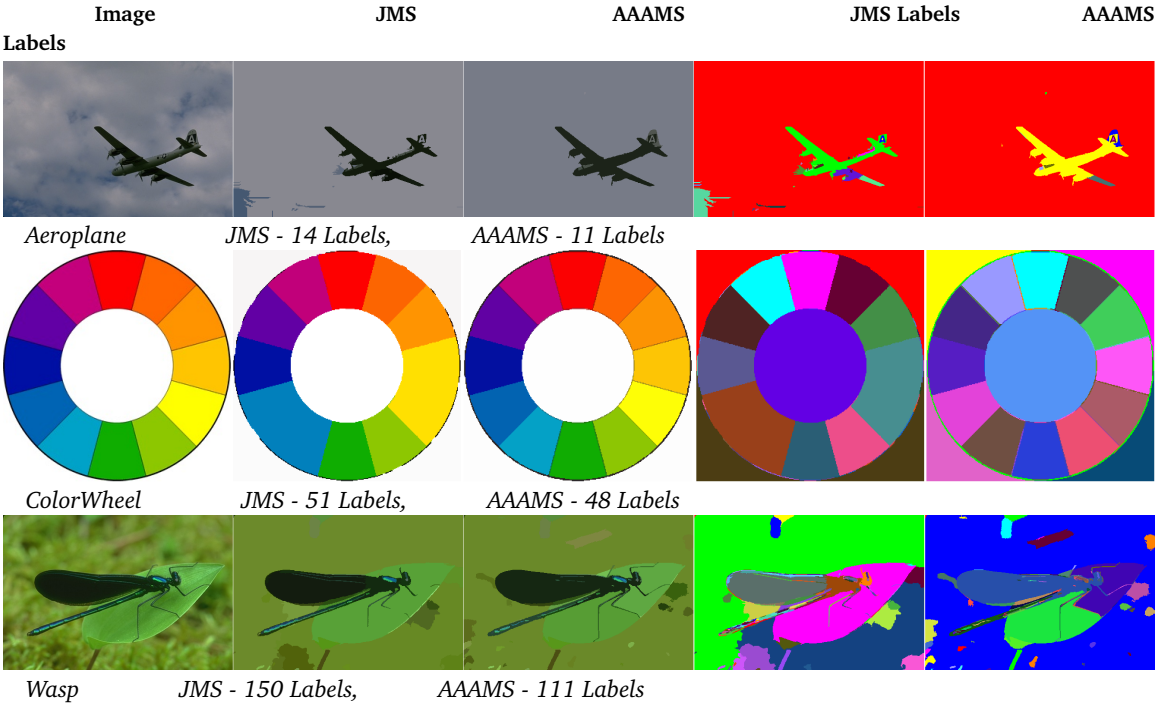


Figure 2.5: **Additional Comparisons :** More parsimonious segmentations were quite often not achievable with JMS - some varied examples are shown above (Images such as *Lady* in *Figure* 2.4 are a typical case too). Both methods were configured for reduced label usage. Minimum cluster size was 10. JMS, at its limit, is breaking boundaries and under segmenting. AAAMS with lesser labels, does not break boundaries, still maintains segment saliency.

Shift implementation (JMS) from EDISON ([50]), over Berkely Segmentation Dataset ([49], BSD300). BSD300 is meant for supervised algorithms - we simply clubbed the training and test images together. For sake of completeness, prior art on unsupervised image segmentation is also shown in *Table* 2.1. All indicated parameter values for AAAMS and JMS are squared. We did not search for the best performing parameter set for AAAMS, opting for a single low valued set instead. AAAMS performed significantly better than JMS, with results superior to other unsupervised image segmentation methods as well.

Our experiments indicated that low base bandwidths, $\langle \sigma^r_{base}, \sigma^s_{base} \rangle$, performed generally well on a good range of images (*Figure* 2.4). This was due to the presented approach being locally adaptive and anisotropic. At similar clustering levels, AAAMS preserved more details and affected more salient segmentations.

Single kernel AAAMS was tested on images and 2D, 3D gaussian mixtures at varied scales. Results in *Figure* 2.3 are with postprocessing disabled. As can be seen, reasonable local

was because humans tend to utilize much more comprehensive cues, and incorporate object or more holistic level semantics in their segmentations. It was noticed that PRI corresponded better to low level segment saliency than others.

Table 2.2: **Results on higher dimension data :** We show results on real world datasets from [51], with a single kernel. Indicated values are in order of AAAMS / MS / VariableMS ([28]) respectively, with best values in red.

Data $\langle$**#Dims**, **#Classes**$\rangle$	***PRI***	***GCE***	***VoI***
$Seeds \quad \langle 7D, 3\rangle$	.89 / .86 / .87	.17 / .20 / .19	0.85 / 0.98 / 0.93
$Yeast \quad \langle 8D, 10\rangle$	.69 / .61 / .67	.44 / .39 / .47	3.03 / 3.10 / 3.22
$Letters \quad \langle 16D, 26\rangle$	.87 / .86 / .83	.67 / .70 / .62	4.96 / 5.16 / 4.72

bandwidths arise, robustly identifying modes and salient clusters, by adapting according to local structure.

Experiments were conducted with some higher dimension datasets from [51] as well. *Table* 2.2 shows initial results, along with comparisons with single domain standard Mean Shift (MS), and [28]'s isotropic variable bandwidth implementation. Cluster count was kept the same as class count. AAAMS post-processing was disabled. [28] first determines isotropic point bandwidths using the k^{th} nearest neighbor distance heuristic, and subsequently utlizes them in single kernel mean shift iterations. Our experiments with it indicated a lack of clustering control. The datasets were meant for supervised classification, with attributes/feature components at different scales, and having uncorrelated and/or uninformative dimensions. Without any pre-processing (normalizations, component analysis) decent results were attained with a single kernel AAAMS. Note that [28] internally normalizes the data, while AAAMS & MS results are without any normalizations.

Promising results, both qualitative and quantitative, are indicative of the efficacy of the presented approach. We intend to experiment further, especially with different merging schemes and on varied data spaces.

2.4 Conclusion

A generalized methodology for feature space partitioning and mode seeking was presented - leveraging synergism of adaptive, anisotropic Mean Shift and guided agglomeration. Unsupervised adaptation of full anisotropic bandwidths is useful and further enables Mean Shift clustering. We are excited about its prospects on point-normal clouds and video streams.

Our experiments did indicate sparse data to be an issue. This is understandable, as it encumbers cluster growth and bandwidth development, with AAAMS behaving like conventional Mean Shift then. Future work would also focus on alleviating this issue.

CHAPTER 3

GEOMETRIC ASSOCIATION WITH SURFACE PATCHES

A fundamental challenge to sensory processing tasks in perception and robotics is the problem of establishing data associations across viewpoints (as in [52, 53]). It is intrinsic to applications such as motion estimation, SLAM, SfM, localization, loop closure, and several other multi-view tasks pertaining to detection, tracking and segmentation ([54, 52, 55, 56, 57, 58, 59, 60] to refer just a few).

We present a robust, generally applicable solution for data association over 3D point sets. The approach is able to ascertain localized, potentially dense, surface patch associations. This is made possible by leveraging macro scale 3D geometry which, as we show, is highly discriminative.

The popularity of laser scanners, depth and RGB-D sensors has led to widespread use of 3D modality in robotics and perception. Current methodologies for 3D data association though, have significant limitations. They either generate sparse correspondences on feature points, assuming a locally discriminative environment [1]; or use complete point clouds to associate based on some form of nearest neighbors, assuming limited sensor motion (restricted viewpoint changes) or availability of priors.

We present a minimally restrictive scheme — it neither requires a locally discriminative environment, nor assumes restrictions on viewpoint changes or availability of priors. It operates by ascertaining standalone surface patch correspondences between the views (of a scene). Motivated by recent trends in scene understanding literature of utilizing image superpixels due to representational compactness and robustness to noise, we propose an analogous model for surface patches. A purely geometric approach is employed — one which works well over real world depth images or point cloud data acquired from range sensors and 3D scanners.

Ascertaining surface patch (alternatively, 3D superpixel) associations without making assumptions on sensor motion, scene geometry and structure, is indeed difficult (and to our knowledge, unsolved). Superpixel decompositions vary in each view, rendering the correspondence inexact. Besides, superpixels are defined through (and for) homogeneity – they are not uniquely discriminable by design. The problem gets complicated further, when

[1] Real world 3D data has rather high local ambiguity. The problem is significantly aggravated in structural and indoor type settings which tends to be locally smooth and isomorphic. And when data is acquired from commodity sensors, as it is usually high in noise and imperfections.

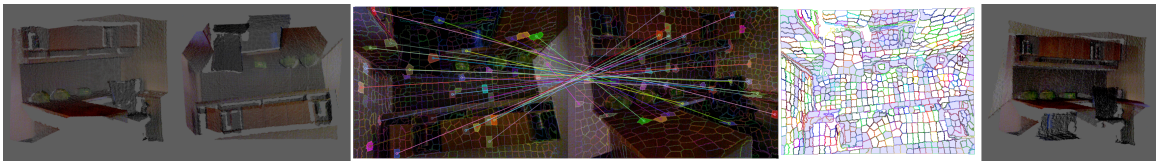


Figure 3.1: **Exemplar patch association result :** Point clouds from two views of a workspace scene are shown on left. The second view was captured with the sensor completely inverted (180° roll), and from a wide baseline. The two views also have significant changes in surface resolution scales, self-occlusions, and changes in yaw & pitch. The image in centre shows a few random samples of surface patch (depth superpixel) associations between the two views, computed using our algorithm. Associated patches are connected by a line and have the same color overlay. The associations were not filtered or post-processed. The centre-right image shows the superpixel decomposition of the second view. The grey overlay over some superpixels indicates the superpixels that are not associated - these include regions which were occluded or absent in the first view. The right-most image shows the unrefined reconstruction obtained directly from the dense superpixel associations. The relative motion/transform was computed simply through corresponding 3D means of the associated superpixels.

appearance information is unavailable altogether.

Nevertheless such associations are very useful - because correspondent superpixels roughly represent the same physical 3D surface patch. As we will show, relative scene geometry over sufficiently large neighborhoods contains adequate discriminative information to achieve potentially dense associations. By regularizing superpixel traits such as surface area and smoothness – the associations can be made to have near co-incident 3D (centroid) localizations as well – affording nice sensor motion estimates even under significant change in perspectives and scant data acquisition (*Figure* 3.1, *Table* 3.2). Importantly, such an approach performs equally well in locally ambiguous (such as isomorphic or textureless) or featureless environments. Furthermore, it can preserve localized semantics (encoded by the superpixel labels) across views. Although not within this article's scope, this enables more straightforward primitive level associations and semantic transfer.

Our methodology is based on invariantly representing a surface patch through a set of relative geometrical properties/features extracted with respect to superpixels in its neighborhood. A uniquely consistent ordering is utilized to sequence this set. Such a representation is invariant to sensor's motion, and is made robust to its noise. Our matching scheme leverages a sequence comparison metric, *Restricted Damerau Levenshtein,* [61] – this pertains to family of sequence alignment algorithms, [62], that have polynomial complexity, are provably optimal, and have been in popular use for large scale sequence matching, especially in bio-informatics community.

As we will show, such a scheme is physically intuitive and is naturally applicable to a geometry matching context. The approach is robust to heavy sensor motion, significant

viewpoint differences, occlusions, partial overlaps and high data noise. It is tolerant of match discrepancies between superpixels, which arise due to varying decomposition across views. The technique also does not require any priors – motion or otherwise, and does not make restrictive assumptions on scene structure and sensor movement. It does not require appearance – is hence more widely applicable than appearance reliant methods, and invulnerable to related ambiguities such as textureless or aliased content. We present promising qualitative and quantitative results under diverse settings, along with comparatives with popular approaches based on range as well as RGB-D data.

We evaluate our approach on ground truth datasets from [63], datasets from [56], and others collected in challenging, yet everyday settings. The experiments are indicative of the efficacy of the proposed approach in computing localized, dense associations. They also indicate more robust performance than popularly used association approaches, based on geometrical and appearance features.

3.1 Related Work

As we are unaware of literature directly addressing our problem setting, we survey some of the relevant work in the broader scope of feature representation and data association / matching, operating over range and RGB-D data. We also refer to some pertinent literature in appearance only settings.

Point cloud association approaches based on local 3D descriptors, like ones used in [64, 65], although useful, can be potentially non-robust. They are hindered in settings which are either locally homogenous or isomorphic, which is not uncommon in everyday scenes and structures. More holistic representations have been used for association - for example, [66] accounts for partial observability of landmarks. Plane representations have been used in dominantly polygonal environments. [67, 68] tentatively associate planes between consecutive frames based on nearest neighbor descriptor matching and relative plane angles respectively, before pruning them through specialized RANSAC schemes. [69, 4] associate by assuming a physical frame to frame overlap between corresponding planes. *Registration approaches* such as [54, 70] require good initialization / restricted motion. [71, 72, 73] present branch and bound schemes for registration, either assuming pure rotations or known correspondences. [74, 75] present globally optimal schemes for aligning object models, utilizing local descriptors and interleaved ICP respectively.

RGB-D based dense approaches like [76, 77, 78] (and [79] which only uses depth) associate based on flow, image warping utilizing photometric errors or ICP alignment, to estimate motion. They afford sensor rotations, but operate under short baseline and under the hypothesis that associations always lie within a neighborhood epsilon. Typically, occlusions

are not handled and temporal consistency is leveraged. Such methods are suitable for settings with constrained motion. [80] utilizes a patch based scheme to track deformable meshes.

RGB-D feature based approaches for more generic SLAM, SfM and motion estimation applications (as in [55, 56, 81, 82]) employ sparse image features (generally SIFT, [83]) back-projected in 3D, to ascertain frame-wise 3D correspondences. [58] augments the geometrical descriptor SHOT, [64], with texture, for improved localization and object detection.

There are *higher level approaches*, which associate by leveraging application specific constraints. [84] utilizes aggregation of densely sampled point features at superpixel levels for RGB-D object detection and recognition. [85] utilizes (color only) superpixel associations to reconstruct piecewise planar scenes under known extrinsics. It assumes similar sensor orientations and imposes restrictions on possible plane orientations. Stereo literature like [86, 87] ascertain disparity maps by associating surfaces / planes relying on short baselines, similar sensor orientations, and discriminative appearance or local features. [59] utilizes planar stereo reconstructs to segment fully observable foreground from multiple views. [88, 89] operate upon SfM point clouds (reconstructed a priori) to reason about visibility and association of planar primitives from multiple views. [90, 91, 92] utilize appearance similarity at patch levels to build a dictionary and associate in the nearest neighbor sense. They find use in image enhancement, and matching scenes with similar appearance.

Matching methodologies, apart from typically assuming availability of discriminative local features, quite often also rely on motion and/or visibility priors - [52] for example, performs exhaustive search over all possible permutations of joint associations (exponential complexity), and attains tractability through priors. Similarly, intractable joint probabilistic formulations used in [93, 53], attain feasibility through priors. Graph techniques have been used to ascertain jointly consistent feature matches. [94] presents a good overview – The approach is to represent features as nodes, with the relative constraints between them as graph edges. An edge preserving mapping between nodes of such graphs is then computed, as either a subgraph isomorphism, or relaxed to inexact graph homomorphism, or as bipartite graph matching problem with non-linear constraints (say, when edges are distances). All of the above formulations are NP-complete and become quickly intractable, especially in absence of priors. Exact matching formulations have been mostly limited to sparse 2D scenarios, as in [95, 96, 97], involving a rather limited number of nodes. [96] utilizes maximum common subgraph formulation to associate sparse 2D laser scans. [5] approximates a dominant solution through eigenanalysis of the graph adjacency matrix. [98, 99] present recent approximate graph based solutions for ascertaining image feature matches – [98] progressively improves skeletal graphs, while [99] employs a density maximization scheme.

3.2 Approach Overview

We start with a regularized patch segmentation (*Section* 3.3). Each superpixel (or the ones of interest) is then expressed through a set of geometric features/relationships arising from all the superpixels in its neighborhood. Each feature in the set corresponds to a superpixel in the considered neighborhood, and is defined through patch level relative geometrical properties expressed invariantly (*Section* 3.2.1). The ascertained feature set, thus, jointly represents all geometrical features of interest in the neighborhood. Finer or coarser geometrical detail can be captured by adjusting the granularity of the superpixel decomposition. Similarly, more global (or local) geometry can be represented by considering larger (or smaller) neighborhoods.

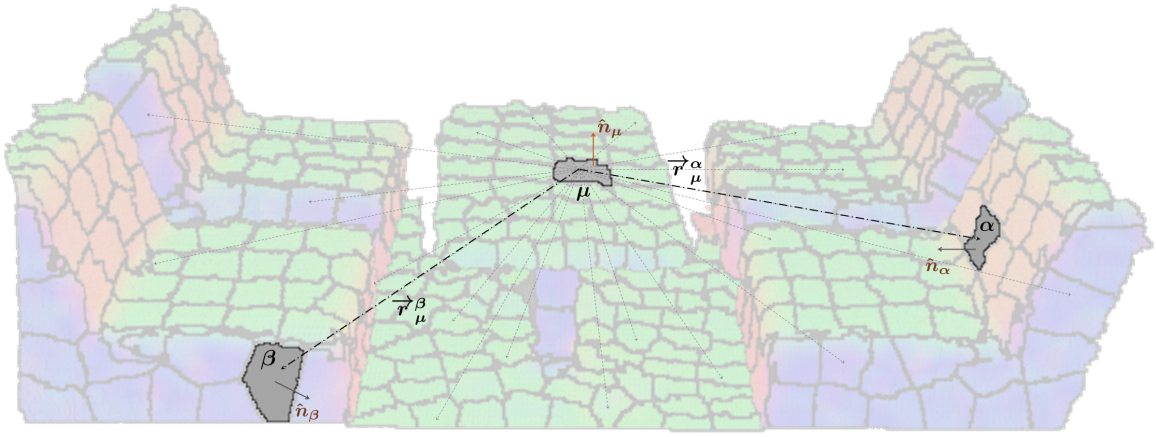
Such a representation captures invariant 3D geometry effectively. It does not require assumptions of the scene structure, such as piecewise or dominant planarity. It is also discriminative enough to disambiguate in difficult settings such as ones with duplicate or locally isomorphic content (*Figure* 3.8).

The feature set of a superpixel is then arranged as a sequence by enforcing an ordering over them (*Section* 3.2.2). The motivation for sequencing is to induce a partial order which remains invariant across views. This is required so that feature sets from different views can be correctly matched.

Our matching scheme (*Section* 3.2.3) utilizes edit distance based sequence comparisons ([62]), specifically the Restricted Damerau Levenshtein distance metric ([61]). The edit distance between two sequences of arbitrary length can be optimally evaluated in quadratic time, and is directly indicative of their dissimilarity. In our context, a feature sequence expresses neighborhood geometry about a given superpixel - with each of its features exclusively capturing geometric information corresponding to a neighboring surface patch. Comparisons between two such sequences, thus, gives us powerful means to ascertain the amount of geometrical mismatch between the two considered neighborhoods[2]. The approach is also inherently robust - as scenarios with partial view overlaps, occlusions and self-occlusions are naturally afforded through the edit operations, and sensor noises can be intuitively accounted for while matching individual features in the sequences (*Table* 3.2 quantifies GASP's performance with increasing baselines, perspective changes and non-overlapping content).

We specify the superpixel decomposition of given a given view of a scene as $S = \{s_i\}_{i=1}^{N}$. We denote μ $(\in S)$ as the superpixel currently under consideration. Similarly, another view

[2]By tractable comparisons, we can now essentially match geometry between 3D neighborhoods in as globalized (or localized) manner, as desired. This is especially useful for cases where geometry about localized/small neighborhoods is not discriminative enough for making associations; large/global neighborhoods need to be considered to disambiguate then.



$$\hat{u}^{\alpha}_{\mu} = \hat{r}^{\alpha}_{\mu}$$

$$\hat{v}^{\alpha}_{\mu} = \frac{\hat{n}_\mu - \left(\hat{n}_\mu \cdot \hat{r}^{\alpha}_{\mu}\right)\hat{r}^{\alpha}_{\mu}}{\left\|\hat{n}_\mu - \left(\hat{n}_\mu \cdot \hat{r}^{\alpha}_{\mu}\right)\hat{r}^{\alpha}_{\mu}\right\|}$$

$$\hat{w}^{\alpha}_{\mu} = \frac{\hat{u}^{\alpha}_{\mu} \times \hat{v}^{\alpha}_{\mu}}{\left\|\hat{u}^{\alpha}_{\mu} \times \hat{v}^{\alpha}_{\mu}\right\|}$$

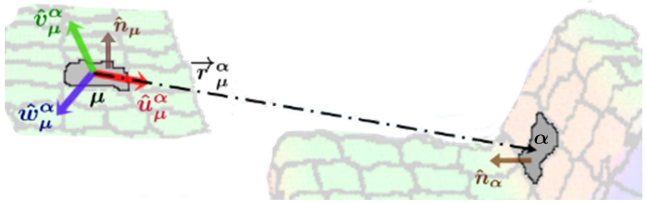


Figure 3.2: **Invariant 3D geometric property extraction :** For a given patch μ, relative and invariant 3D geometric properties are extracted with respect to other patches in a non-local neighborhood. To facilitate that, an orthonormal frame agnostic to the sensing viewpoint is derived using the Gram-Schmidt process.

of the scene will have a decomposition $S' = \{s'_j\}_{j=1}^{N'}$. μ' would denote a superpixel from S' currently being considered for possible correspondence with μ. $\aleph_\mu$ indicates the set of nearest superpixels in μ's 3D neighborhood, with $|\aleph_\mu|$ indicating the cardinality of the set. α refers to an arbitrary superpixel in μ's neighborhood ($\alpha \in \aleph_\mu$).

3.2.1 Capturing 3D Geometry

We utilize $\aleph_\mu$, to express μ through a set of transformation invariant geometrical features, $Q_\mu = \{q^{\alpha}_{\mu}\}_{\forall \alpha \in \aleph_\mu}$. Each superpixel, α, in the neighborhood, $\aleph_\mu$, contributes geometric information, q^{α}_{μ} , and helps capture the geometry in μ's 3D neighborhood. Let $\hat{n}_\mu$ indicate μ's surface normal, and l_μ indicate its 3D location. $\overrightarrow{r}^{\alpha}_{\mu} = l_\alpha - l_\mu$, would indicate the relative displacement of the superpixel; $\hat{r}^{\alpha}_{\mu}$ and $||\overrightarrow{r}^{\alpha}_{\mu}||$ would indicate its direction and magnitude respectively. Evidently, the quantities $\hat{n}_\mu$, $\hat{n}_\alpha$, $\overrightarrow{r}^{\alpha}_{\mu}$ depend on the reference frame. In order to make them invariant to the sensor pose, we express them in a coordinate frame derived from superpixels μ and α themselves. *Figure* 3.2 illustrates this. It also shows how an orthonormal co-ordinate frame is derived from $\hat{n}_\mu$ and $\overrightarrow{r}^{\alpha}_{\mu}$ using the Gram - Schmidt process.

$\hat{u}^{\alpha}_{\mu}$, $\hat{v}^{\alpha}_{\mu}$, w^{α}_{μ} form the orthonormal basis. This basis is almost never degenerate, as $\hat{n}_\mu$ and $\overrightarrow{r}^{\alpha}_{\mu}$

are rarely co-linear. $\hat{n}_\alpha$ can now be expressed in this local frame through the projection components $[\hat{n}_\alpha \cdot \hat{u}_\mu^\alpha,\ \hat{n}_\alpha \cdot \hat{v}_\mu^\alpha,\ \hat{n}_\alpha \cdot \hat{w}_\mu^\alpha]^T$. For two given superpixels, μ and α, these components would remain independent of the sensor viewing frame. Additional pieces of relative, invariant information can be extracted through $\hat{n}_\mu$, $\hat{n}_\alpha$ and $\overrightarrow{r}_\mu^\alpha$ as follows

$$\theta_{\alpha,\mu} = cos^{-1}\left(\hat{n}_\mu \cdot \hat{n}_\alpha\right) \tag{3.1a}$$

$$\theta_{r,\mu} = cos^{-1}\left(\hat{r}_\mu^\alpha \cdot \hat{n}_\mu\right) \tag{3.1b}$$

$$\theta_{r,\alpha} = cos^{-1}\left(\hat{r}_\mu^\alpha \cdot \hat{n}_\alpha\right) \tag{3.1c}$$

q_μ^α can now be expressed as a feature vector constituting of seven relative, invariant elements. We have then

$$\begin{aligned} q_\mu^\alpha = [\ ||\overrightarrow{r}_\mu^\alpha|| \cdot sgn(\hat{n}_\alpha \cdot \hat{u}_\mu^\alpha),\ ||\overrightarrow{r}_\mu^\alpha|| \cdot sgn(\hat{n}_\alpha \cdot \hat{v}_\mu^\alpha), \dots \\ \dots ||\overrightarrow{r}_\mu^\alpha|| \cdot sgn(\hat{n}_\alpha \cdot \hat{w}_\mu^\alpha),\ ||\overrightarrow{r}_\mu^\alpha||,\ \theta_{\alpha,\mu},\ \theta_{r,\mu},\ \theta_{r,\alpha}\ ]^T \end{aligned} \tag{3.2}$$

where $||\overrightarrow{r}_\mu^\alpha||$ is included for additional redundancy. We utilize the signs of $\hat{n}_\alpha$'s projection components (with $sgn(.) \in \{-1, 0, 1\}$), as their actual values tend to be noisy. A proper approach is to utilize an epsilon-insensitive signum function, for example $sgn_e(.)$ rather than $sgn(.)$ - it is defined as zero either when $1)$ the angle between $\hat{n}_\alpha$ and the respective basis vector $\notin [e_\theta,\ PI - e_\theta]$; or in the uncommon case of degenerate basis when $2)$ $\hat{n}_\mu$ is co-linear with $\overrightarrow{r}_\mu^\alpha$ (that is, when $\theta_{r,\mu} \notin [e_\theta,\ PI - e_\theta]$). e_θ is the allowable angular noise tolerance. The component signs are then scaled by $||\overrightarrow{r}_\mu^\alpha||$ in order to incorporate signed distance information between μ and α. The feature, q_μ^α, is thus expressed stably in presence of noises. It captures relative information between μ and the neighborhood superpixel α – the relative pose, distance, orientation and bearings. Understandably, the elements of q_μ^α would be affected by noise. However, our matching methodology is robust to it, explicitly accounts for it (*Section* 3.2.3, deviation thresholds). The joint feature set $Q_\mu = \{q_\mu^\alpha\}_{\forall \alpha \in \aleph_\mu}$, constituting of relative feature vectors from all superpixels in $\aleph_\mu$, thus, essentially expresses the geometry in μ's neighborhood invariantly.

3.2.2 Uniquely Consistent Partial Ordering

Once the geometric feature set has been determined, a partial ordering needs to be imposed on its elements to obtain a feature sequence - as our matching scheme leverages a distance metric based on sequence aligning comparisons. An ordering over μ's neighborhood $\aleph_\mu$, can be denoted as $\overleftarrow{O}_\mu = \langle a, b, c \dots \alpha \dots \rangle\ , \forall \alpha \in \aleph_\mu$ – where $\langle a, b, c \dots \rangle$ indicates the ordered sequence of superpixels. This is used to order the joint feature set Q_μ as

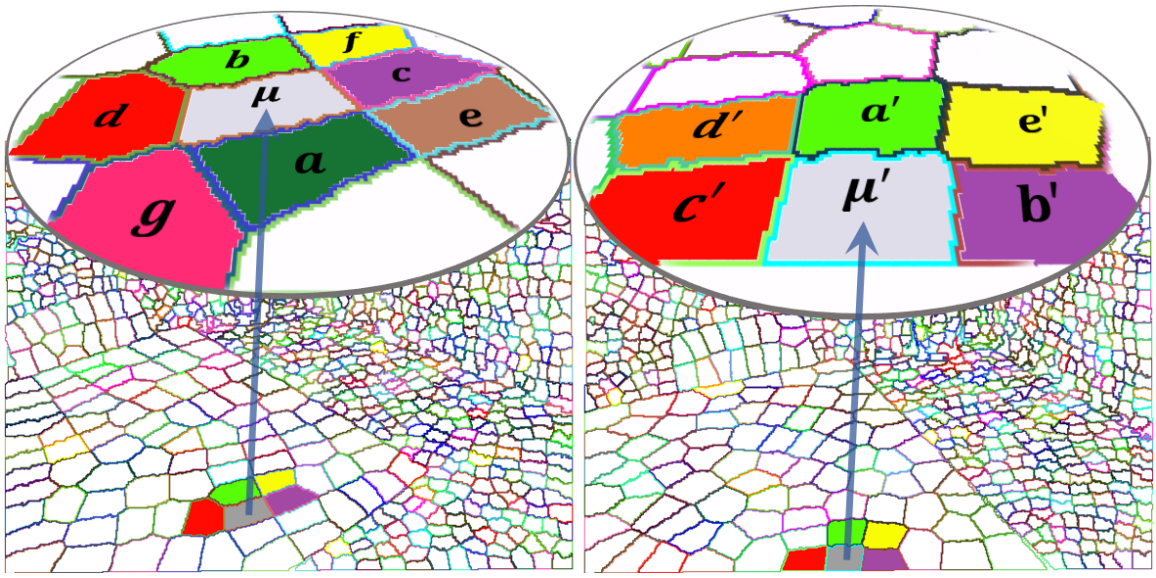


Figure 3.3: **Mutually consistent orderings :** Illustrative consistent orderings of immediate neighborhoods of associating superpixels, μ and μ' are shown. The orderings are indicated by alphabetical progression of the marked neighboring superpixels. The matching pairs of superpixels are shown on the table, and share a common color. The orderings are consistent as the sets of corresponding neighborhood superpixels {***b***, ***c***, ***d***, ***f***} and {***a'***, ***b'***, ***c'***, ***e'***}, as indicated by their increasing alphabetic order, arise identically in the orderings.

$\overleftarrow{Q}_\mu = \langle q_\mu^a, q_\mu^b, q_\mu^c \dots q_\mu^\alpha \dots \rangle$. These ordered sequences are subsequently utilized to ascertain a potential association between two given superpixels, $\mu \,\&\, \mu'$ from different views. The approach is to devise an ordering scheme such that two superpixel orderings, $\overleftarrow{O}_\mu \,\&\, \overleftarrow{O}_{\mu'}$ (over $\aleph_\mu \,\&\, \aleph_{\mu'}$) are both partially ordered by (with respect to) their matching subsequences - these subsequences would be the identically ordered (sub-)sets of corresponding superpixels in neighborhoods of μ and μ' respectively. To put it simply, *the order of the correctly corresponding superpixels in the neighborhoods* $\aleph_\mu$ *and* $\aleph_{\mu'}$ *respectively, should remain invariant in the respective orderings,* $\overleftarrow{O}_\mu$ and $\overleftarrow{O}_{\mu'}$ ($\overleftarrow{O}_\mu$ and $\overleftarrow{O}_{\mu'}$ should be consistent). *Figure* 3.3 illustrates consistent orderings, $\overleftarrow{O}_\mu =< a, b, c, d, e, f, g >$ and $\overleftarrow{O}_{\mu'} =< a', b', c', d', e' >$, over immediate neighborhoods of two correctly associating superpixels μ and μ'. These orderings are *mutually* consistent as the correct correspondences in the neighborhoods form subsequences – $< b, c, d, f >$ and $< a', b', c', e' >$ arise in identical order in $\overleftarrow{O}_\mu$ and $\overleftarrow{O}_{\mu'}$ respectively.

To achieve ordering consistency, we utilize Q_μ itself which already constitutes of a superpixel-wise set of invariant geometric features relative to μ. In effect, $\overleftarrow{Q}_\mu$ is simply ascertained through a robust sorting operation over Q_μ's elements ($\{q_\mu^\alpha\}$, which are invariant). Note that since this ordering is derived from geometry with respect to μ (*Figure* 3.2, Equation 3.1), it will not, in general, be consistent with an ordering over an arbitrary superpixel from S'. It will only be consistent with an ordering, say $\overleftarrow{O}_{\mu'}$, defined about some superpixel μ' in S', which

has similar relative geometry in its neighborhood as μ – which is precisely the objective[3]. Thus $\overleftarrow{Q_\mu} = Sort(Q_\mu, e_r, e_\theta)$; with the sorter conducting pairwise comparisons between Q_μ's constitutent features. The second dimension (of the two features being compared) is only used if the first dimension is equivalent, the third dimension is only used if the first two are equivalent, and so forth. Equivalence is defined as the values being within epsilon tolerances of each other, to affect resolution, and account for noise and finite precision numerical errors. A distance tolerance, e_r is used, along with the afore-utlilized angular tolerance, e_θ. In our experiments over noisy Kinect data, $e_r = .02$ metres and $e_\theta = {}^{5\pi}/_{180}$ radians, worked well.

3.2.3 Matching Patches

A pair of superpixels, μ and μ', can now be compared for geometric correspondence using their respective ordered feature sequences $\overleftarrow{Q_\mu}$ and $\overleftarrow{Q_{\mu'}}$. We utilize a sequence matching scheme based on [61], to ascertain/quantify the dissimilarity between the sequences in the form of edit distances. Edit distance based algorithms, [62], operate by editing one sequence into another. By utilizing efficient dynamic programming, they progressively compare two elements at a time, one from each sequence. If the elements match up, the next pair of elements is considered – else element in one of the sequences is edited first at a cost (typically by inserting, deleting or replacing it), before resuming the comparisons. Sequences which have matching elements will result in lower edit distances than ones which do not. Additionally, sequences which have matching elements in the same order will result in lower edit distances than ones which do not. The computed distances for algorithms such as [61] are optimal with respect to the specified editing costs.

Two features, q_μ^α and $q_{\mu'}^{\alpha'}$, from the respective sequences $\overleftarrow{Q_\mu}$ and $\overleftarrow{Q_{\mu'}}$, will match up when relative geometry of patch α with respect to μ , is the same as the relative geometry of α' with respect to μ' (q_μ^α and $q_{\mu'}^{\alpha'}$ would then hold approximately same values). If μ and μ' have the same relative geometry in their neighborhoods, the sequences on the whole will naturally match, and will not require many edit operations – resulting in low edit distances; else the distances will be high. Additionally, by devising an ordering utilizing the unique relative features themselves, two sequences which do not capture similar geometry will match up badly, because their ordering will differ significantly. We utilize the *Restricted Damerau–Levenshtein* (RDL, [61]) algorithm for ascertaining sequence disparity. In contrast to the popular Levenshtein algorithm ([62]), which allows insert, delete and replace operations over sequence elements, RDL allows the additional operation of transposition of adjacent elements[4] , and has the additional constraint that each subsequence can be edited only once.

[3] $\overleftarrow{O_\mu}$ will, in fact, be quite inconsistent with an ordering about a non-corresponding superpixel in S' and hence, as a consequence of mutually inconsistent orderings, will result in rather poor match distance.

[4] Assuming all edit weights to be unity, the Levenshtein distance between string sequences, $ABCD$ & BAC, is 3, while the Restricted Damerau-Levenshtein distance is 2 – due to an aligning transposition.

Algorithm 3.1: *Restricted Damerau Levenshtein* [$\boldsymbol{CompareRDL}$]

Function `CompareRDL`

Input : $|\overleftarrow{Q_\mu}|$, $|\overleftarrow{Q_{\mu'}}|$, $< Input - Parameters >$

Output : $tab[L_\mu, L'_\mu]$

Input-Parameters : $insert \leftarrow 1$, $delete \leftarrow 1$, $replace \leftarrow \infty$, $switch \leftarrow 0$

$L_\mu \leftarrow |\overleftarrow{Q_\mu}|$

$L'_\mu \leftarrow |\overleftarrow{Q_{\mu'}}|$

$tab\ [1,1] \leftarrow 0$

foreach $i \in [2, L_\mu]$

  $tab[i,1] \leftarrow tab[i-1,1] + insert$

foreach $j \in [2, L'_\mu]$

  $tab[1,j] \leftarrow tab[1,j-1] + delete$

foreach $j \in [2, L'_\mu]$

  foreach $i \in [2, L_\mu]$

    If `Match`$(\overleftarrow{Q_\mu}\ (i), \overleftarrow{Q'_\mu}\ (j))$

      $substitute \leftarrow 0$

    Else

      $substitute \leftarrow replace$

    $$tab[i,j] \leftarrow min\left\{ \begin{array}{c} tab[i-1,j] + insert, \\ tab[i,j-1] + delete, \\ tab[i,j-1] + substitute \end{array} \right\}$$

    If $i > 2$ & $j > 2$ & `Match`$(\overleftarrow{Q_\mu}\ (i-1), \overleftarrow{Q'_\mu}\ (j-1))$

      $tab[i,j] \leftarrow min(tab[i,j], tab[i-2,j-2] + switch)$

End

Function `Match`

Input : q^β , q^γ , $< Input - Parameters >$

Output : $\{True\ or\ False\}$

Input-Parameters : $r_{dev} \leftarrow UserDefined$, $\theta_{dev} \leftarrow UserDefined;$, $replace \leftarrow \infty$, $switch \leftarrow 0$

$q_{noise} \leftarrow [\theta_{dev}, \theta_{dev}, \theta_{dev}, r_{dev}, r_{dev}, r_{dev}, r_{dev}]$

$\Delta q \leftarrow abs(q^\beta - q^\gamma)$

foreach $t \in [1, 7]$

  If $\Delta q[t] > q_{noise}[t]$

    return $False$

return $True$

End

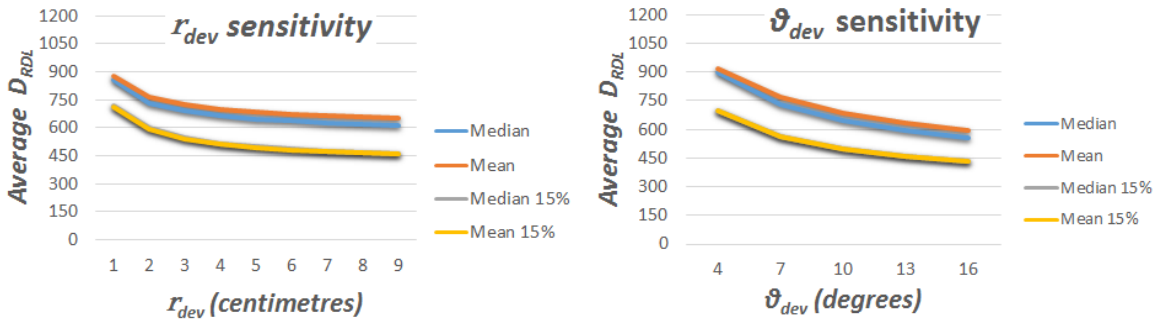


Figure 3.4: **Match threshold sensitivity :** Impact of varying match thresholds r_{dev} & θ_{dev} on averaged $D_{\mu_{RDL}}$ is shown. Default r_{dev}, θ_{dev} were 5 cm and $10°$. Mean and Median edit distances, over all associations, and over top 15% are shown.

The edit operations' costs can be set arbitrarily to suit a use case, and to achieve a desired resolution. Comparing two sequences through RDL is a symmetric operation, and sequences of different sizes can be compared.

These properties suit our needs nicely. Superpixels in the neighborhood $\aleph_\mu$ may not be present in $\aleph_{\mu'}$ and vice versa. This would be because of partial overlap of content between views, and because of occlusions. The operations of insert and delete will basically edit such non-matching features from the sequences. The ability to transpose adjacent features accounts for slight errors in the sequence orderings[5]. Replacement in this context, being physically meaningless, is disabled. Also, the restriction that each subsequence can be altered only once, prevents any re-edits over incumbent feature alignments. Insertion, deletion are symmetric operations and we nominally set their cost to unity. Transposition cost is set to zero, as it only occurs only due to slight ordering inconsistencies.

Obtaining a match between two given features, q_μ^α and $q_{\mu'}^{\alpha'}$, while comparing $\overleftarrow{Q_\mu}$ and $\overleftarrow{Q_{\mu'}}$ is easy. Basically, a match is established when the respective components of the two features lie within some acceptable range of each other. Two simple, intuitive thresholds – one for allowable angular deviation, θ_{dev} and the other for allowable distance deviation, r_{dev} – are utilized. These thresholds account for noise and allowable slack in elements of q_μ^α and $q_{\mu'}^{\alpha'}$. *Figure* 3.4 plots the effect of varying them on (average) edit distances. Also, more precise, co-incident localizations can be achieved by considering smaller values, and a more granular superpixel discretization (vice versa is applicable too).

Two superpixel sequences from different views corresponding to the same 3D location, should have zero edit costs – assuming complete overlap of neighborhoods, no occlusions and consistent segmentations. In practice, correctly associated superpixels (their sequences) would still have some edit costs due to partially overlapping neighborhood geometry, occluded regions, and inexact superpixel decomposition across views. Desirably, these absent,

[5]Our experiments indicated that, in less noisy settings/good datasets, the transposition operation could be optionally disabled without significant impact on association accuracies, due to the use of epsilon tolerances.

occluded, or mismatched superpixel features would be edited out. Incorrect associations will have significantly higher edit distances, as a consequence of dissimilar neighborhood geometry. *Algorithm* 3.1 specifies the matching algorithm. It returns the net edit cost, $D_{RDL}(\mu, \mu')$, between the feature sequences of μ and μ', being considered for association. For clarity, embellishments for memory and computational efficiency have been left out. Note that $D_{RDL}(\mu, \mu') \equiv D_{RDL}(\mu', \mu)$.

3.2.4 Ascertaining Associations

The best potential, putative association for a patch, $\mu \in S$, would be the patch in S' whose neighborhood geometry matches most with μ's neighborhood - one whose feature sequence gives the lowest edit distance with $\overleftarrow{Q}_\mu$.

$$D_{\mu_{RDL}} = min_{\forall \alpha' \in S'}(D_{RDL}(\mu, \alpha')) \tag{3.3a}$$

$$\mu'_{best} = \underset{\forall \alpha' \in S'}{\arg\min}(D_{RDL}(\mu, \alpha')) \tag{3.3b}$$

where $D_{\mu_{RDL}}$ indicates the edit distance from the best association in S', μ'_{best}. $D_{\mu_{RDL}}$ is basically indicative of the amount of rigid geometry mismatch between the neighborhoods of μ and its best putative association; and can hence be utilized to ascertain whether the putative association is considered correct. Normalized edit distances are used for this purpose [6]. For view to view matching, all patches in a view can be made to use equal size neighborhoods (that is, the set of nearest patches in 3D) ; thus $|\aleph_\mu| = k_S,\ \forall \mu \in S$ and $\aleph_{\mu'}| = k_{S'},\ \forall \mu' \in S'$. $D_{RDL}(\mu, \mu')$, for any μ and μ', can thus have a maximum value of $(k_S + k_{S'})$, assuming unit costs for insert/delete operations. Normalized edit distance is then obtained as

$$\hat{D}_{\mu_{RDL}} = \frac{min_{\forall \alpha' \in S'}(D_{RDL}(\mu, \alpha'))}{(k_S + k_{S'})} \tag{3.4}$$

$D_{\mu_{RDL}}/\hat{D}_{\mu_{RDL}}$ are dependable measures of association quality. A putative association for a given patch, μ, is considered correct in the geometric sense, if $\hat{D}_{\mu_{RDL}}$ is not more than a given normal gating value, $\lambda \in [0, 1]$ ($\hat{D}_{\mu_{RDL}} \leq \lambda$ for association). A lower λ would result in more confident and localized associations, while denser but possibly coarser associations would arise at higher λ gatings [7] (depending on a scene's geometrical ambiguity, considered

[6] With pertinent application specific adjustments, normalized edit distances are amenable to a probabilistic interpretation as well.

[7] Alternatively, or when more accurate metric transforms is the end objective, a dense set of putative associations could be first obtained using a high gating; these could be subsequently filtered on the basis of 3D

Table 3.1: **Feature set means are discriminative :** Averaged percentage of best associations with increasing query sizes.

# Feature means queried - C	25	50	75	100
$\mu'_{\mathbf{best}}$ ***found (% Avg.,*** $\lambda = .5$***)***	71.5	86.8	94.5	98.9

neighborhood sizes and match deviation thresholds).

Obtaining a putative association, when comparing with all patches in S', would have a worst case complexity of $O(|S'|k_S k_{S'})$. Although the cubic complexity is tractable, significant further improvements are possible. Some discriminative information can be leveraged from the feature set, Q_μ's mean, $\overline{Q}_\mu$. If two patches, μ and μ' form a correct correspondence, their respective feature set averages, $\overline{Q}_\mu$ and $\overline{Q}_{\mu'}$ would be close to each other. To find the putative match for μ in S', we therefore build a KD-tree over feature set means (normalized) of all patches in S', and search/query for C of the nearest neighboring feature set means to $\overline{Q}_\mu$. The putative association, and subsequently a possible correct association for μ, is then ascertained from the patches corresponding to these queried feature means rather than considering all patches in S'. This brings down the complexity of ascertaining a putative association to quadratic – $O(Ck_S k_{S'})$, where C is the constant number of queries, with $C \ll S'$. *Table* 3.1 indicates the average percentage of best associations (μ'_{best}) found, as a function of query size, C – at least an order of magnitude reduction in computations is achieved, without any significant impact on association accuracies. An early termination criteria in *Algorithm* 3.1 gives another significant improvement. Since associations with normalized distances above λ are anyways ignored, *Algorithm* 3.1 can be terminated prematurely as soon as the edit costs exceed $\lambda(k_S + k_{S'})$. This is generally the case for a majority of potential associations in S', and results in significant gains in practice. Further gains are possible, like screening of possible associations before computing D_{RDL}, utilizing progressive rigid transform estimates. Also note that the associations are computable in parallel — such kinds of efficiency gains is a subject of ensuing work.

Coarse to fine association **:** Significant efficiency gains can also be achieved by ascertaining patch associations in a coarse to fine fashion. Starting with a surface patch segmentation hierarchy, patch associations are ascertained first at the coarsest level, which has a significantly reduced number of patches and is hence faster. The 3D euclidean transform evaluated from patch associations at this coarse level can then serve as a strong prior for associating at a finer level, since it is quite accurate by itself (*Figures* 3.5 and 3.9). While ascertaining patch associations at each of the finer levels, we then utilize the incumbent transform estimate (evaluated from a coarser level) to filter out all but the most probable patch associations. This can be accomplished by first transforming a patch at that hierarchical level in S, say $\mu_h \in S$,

rigid transformation consistency, using a scheme such as RANSAC (*Section* 3.4). Similarly, for a semantic transfer / segmentation task, associations obtained with a high gating could be smoothed out, in a framework such as Conditional Random Fields.

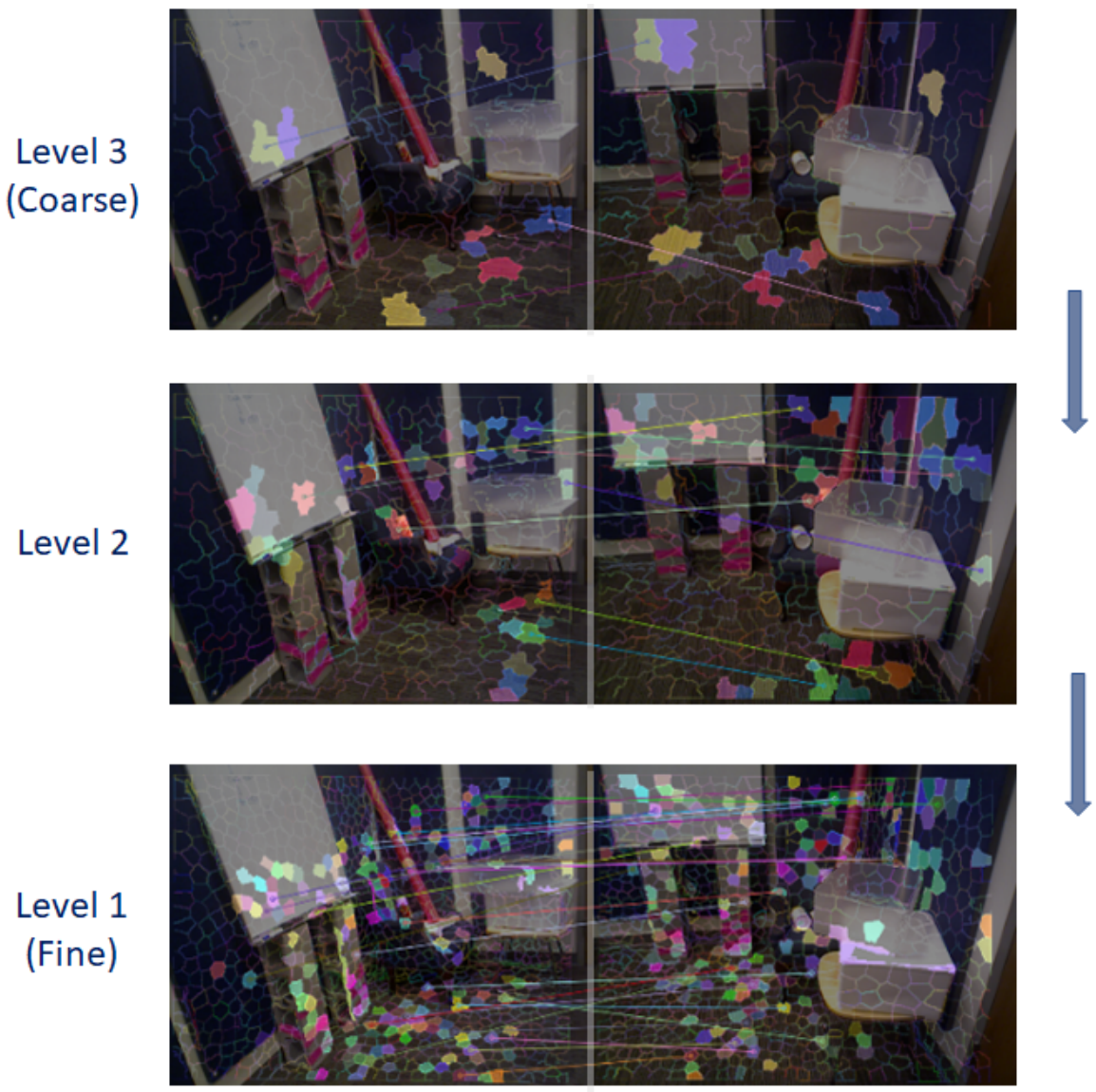


Figure 3.5: **Coarse to fine GASP :** A coarse to fine association example, operating over 3 levels of patch segmentation hierarchy is shown. The scene's views (left and right images) have been shown in color for better illustration, although appearance was not used at all. The patches at the coarsest level, Level 3, are matched first. A few sample patch correspondences have been indicated by overlays of common color and some connecting lines. The transform estimated from matches at the coarsest level are utilized to significantly prune down the set of potential matches for patches in the level below. Note that the correspondences at the coarsest level are well localized as well, despite the steep change in viewpoint and accompanying challenges.

with the incumbent transform estimate. $\mu'_{h,best}$ can then be evaluated by only considering a few patches in S' (at the same level in hierarchy) that are nearest in the euclidean sense to the transformed μ_h. A coarse to fine association example is shown in *Figure* 3.5. Such an approach results in a significantly more efficient scheme overall. Desirably, this allows us

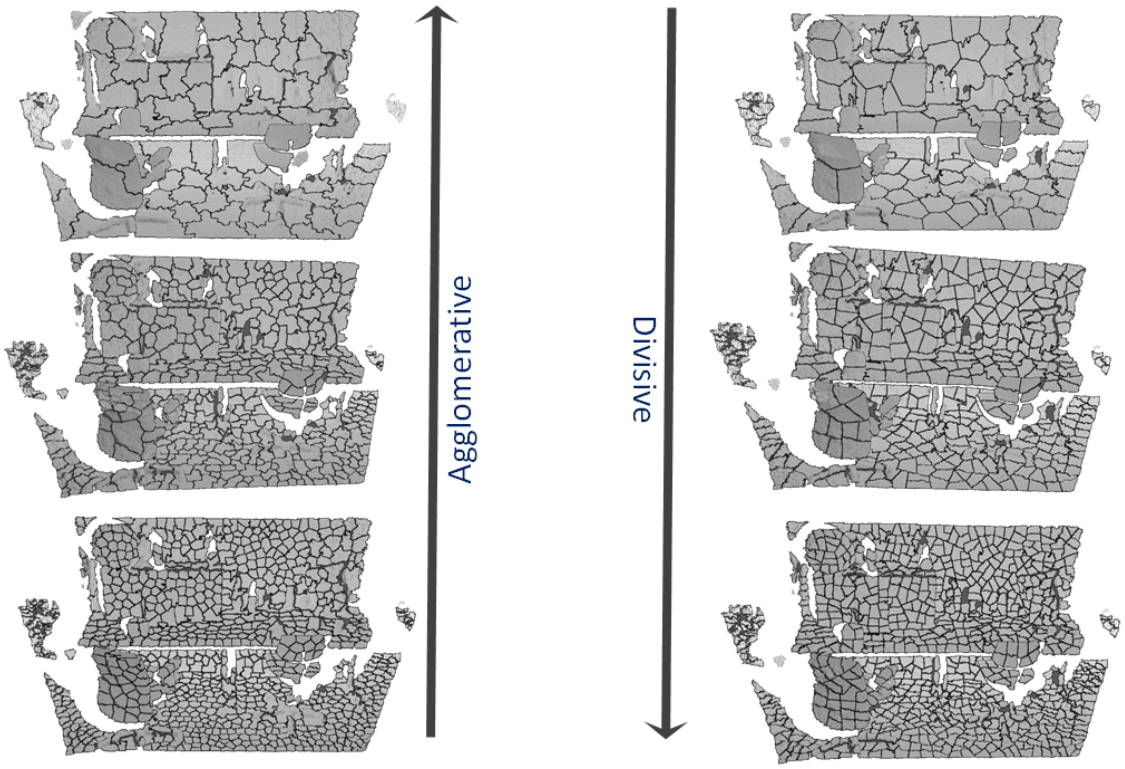


Figure 3.6: **Generating a segmentation hierarchy :** Example patch segmentation hierarchies are shown on left and right. The left hierarchy was generated bottom-up, by agglomerating 3D adjacent patches, starting with the segmentation at the bottom. The hierarchy on the right was generated in a top-down fashion, by subdividing each patch into smaller ones, starting with the segmentation at the top. Note that the divisive scheme preserves surface boundaries better.

to trade-off some accuracy to increase efficiency further as well — as *Figure* 3.9 indicates, the transform estimates resulting from limiting association to the coarser levels are quite accurate as well.

3.3 Patch Decomposition

The methodology presented here can work with any 3D patch segmentation scheme — as long as it generates compact, geometrically regularized 3D superpixels of similar area. The later property helps in achieving patch correspondences that are better localized and spread more uniformly across the (overlapping) volume of the scene.

Such patch decompositions can be obtained from 3D point cloud as well as range data. For example, [100], which segments volumetrically, could be used while working with point clouds; and a surface segmentation scheme such as one presented in [101], or in *Section* 3.3.1

could be employed when working with depth / range images. Besides experimenting with the segmentation scheme outlined in *Section* 3.3.1, we have also experimented with the pointcloud segmentation approach presented in [100]. Both performed well in our experiments.

***Generating a segmentation hierarchy* :** Starting with a patch decomposition at the base layer, a hierarchy of segmentations can then be generated in either fine to coarse (agglomerative), or coarse to fine (divisive) fashion.

In case of fine to coarse, each progressively coarser segmentation layer in the hierarchy is built from local agglomeration of surface patches in the previous layer. This is done by running K-Means in 3D over the centroids of the surface patches in the previous layer. A separate K-means process is run for each set of patches that delineate a smooth surface component, and near uniform seeding is used. In case of coarse to fine, each progressively finer segmentation layer in the hierarchy is built by subdividing the patches in the previous layer. This is again achieved by running K-Means in 3D with uniform seeding, albeit over the constituent points of each patch. The number of clusters are reduced (for agglomerative) or increased (for divisive) by a similar factor at every level. It is ensured that the generated superpixels constitute of points that are intra-connected in 3D, and any superpixels below a certain size are merged into an adjacent one. *Figure* 3.6 shows examples of both divisive and agglomerative hierarchical segmentation. Note that the divisive scheme allows better preservation of surface boundaries through the hierarchy, as any patches in a generated level would always conform to the boundaries of the parent patch.

3.3.1 Depth image segmentation

RGB-D Scene
Only depth used)

3D Normal Map

3D Components

Surface Patches
HexGrid Regularized

Surface Patches
Area Regularized

Figure 3.7: **A depth image segmentation example :** The identified surface components in 3D are indicated by the center image. The images on the right indicate different patch regularization schemes. The hexgrid regularization was obtained by simply introducing a hexagonally tiled label image in f_{cmp}.

We present a depth image segmentation approach that essentially involves decomposing each contiguous, 3D surface (not necessarily planar), into compact (not necessarily small) smooth patches of similar surface area. The resultant patch segmentations are thus consistent across views and are uniform in 3D

Our patch segmentation algorithm takes in a range or depth image, P, and corresponding

point normals, N_p, as input. It also takes in a boolean comparator function, f_{cmp}, which checks two points for 3D connectivity and local smoothness. The output is a label vector L_p indicating membership of each point to some superpixel $s_i\,(\in S$ - where S indicates the set of superpixels), and Φ_s which refers to the collective set of superpixel properties (normal, centroid, area, size). The algorithm also outputs a superpixel connectivity graph, G_s^c, which maintains the 3D adjacencies between superpixels / nodes in G_s^c. We summarize the algorithm below - its worst case complexity is $\mathcal{O}(|P|\log|P|)$.

- *Build a triangulated mesh, M_p and a 3D point connectivity graph, G_p^c* $\rightarrow$ Traverse the points in P for a) ascertaining 3D connectivity of each point with its grid adjacent points, using f_{cmp} for pairwise comparisons and, b) adaptively triangulating each point with its grid adjacent and 3D connected neighbors. This results in a 3D point connectivity graph, G_p^c and a triangulated mesh, M_p. Each triangle primitive in M_p has an associated surface area.

- *Evaluate connected components from G_p^c* $\rightarrow$ Use a breadth first traversal over G_p^c to identify connected components. Each component represents a smooth 3D surface, or an unregularized superpixel.

- *(optional) Evaluate surface areas and decompose components into surface patches* $\rightarrow$ Traverse through the triangle primitives in M_p to populate the surface area of each component. Determine the number of patches for each component (ratio of component surface area with desired area). For each component, cluster the 3D points compactly into the required number of patches, using minimum variance hierarchical K-Means with seeds that are spread uniformly in 3D.

- *Build L_p and Φ_s* $\rightarrow$ Populate L_p, which indicates membership of each point to some superpixel. And agglomerate first and second order statistics for each superpixel, for evaluating Φ_s.

- *Build G_s^c* $\rightarrow$ Traverse the pixel connectivity graph, G_p^c, to build a superpixel connectivity graph, G_s^c.

- *Merge back* $\rightarrow$ Visit nodes (superpixels) of G_s^c in order of increasing size. Using edge contraction operations, merge superpixels below a certain size to adjacent superpixels with the most aligned normal (or closest centroid). During each merge, update superpixel memberships in L_p, and statistics for Φ_s.

f_{cmp} is defined as a pairwise comparator which ascertains compatibility between two adjacent points. For segmenting depth images, it is purely geometric, ascertaining local smoothness

Table 3.2: **Quantitative evaluations and comparisons :** We demonstrate the localization accuracy of GASP's superpixel associations by utlizing them for motion estimates, over kinect datasets from [63] which have ground truths obtained from a motion-capture system. Translation & Rotation RMS errors and failure rates are shown. For all metrics, lower values are better. As can be seen, the transform estimates from GASP associations are accurate. They remain consistent under increasing frame skips, and with minimal failures. We also compare with geometric as well as appearance based 3D feature approaches (ones below short solid lines), in popular use today. Top values are ordered as rgb. GASP performed best overall.

Datasets [63]		*Cabinet – SparseStructure*				*Structure – NoTexture*				*Household – Clutter*		
Frame Skip		10	20	30	40	25	50	75	100	10	50	100
Trans (mtr)	*GASP*	**0.057**	**0.075**	**0.070**	**0.073**	**0.026**	**0.037**	**0.038**	**0.039**	0.023	**0.028**	**0.058**
	SHOT	0.184	0.281	0.352	0.461	0.132	0.228	0.309	0.461	0.033	0.191	0.347
	FPFH	0.185	0.424	0.393	0.435	0.164	0.202	0.275	0.406	0.045	0.260	0.606
	C–SHOT	0.157	0.255	0.363	0.487	0.100	0.213	0.300	0.362	0.030	0.061	0.236
	SIFT	0.201	0.285	0.405	0.468	0.046	0.059	0.098	0.198	**0.014**	0.029	0.095
	D–SIFT	0.131	0.178	0.246	0.242	0.030	0.042	0.191	0.182	0.019	0.138	0.425
Rot (deg)	*GASP*	**1.656**	**1.971**	**2.737**	**2.596**	**0.802**	**0.998**	**1.157**	**1.186**	1.471	**1.077**	**2.359**
	SHOT	4.582	9.714	9.166	13.195	4.392	7.226	8.860	9.896	1.973	6.235	14.494
	FPFH	5.375	11.375	10.146	11.671	5.334	6.729	7.764	14.748	2.157	8.656	22.387
	C–SHOT	4.359	7.678	9.231	11.666	4.177	7.212	10.571	8.592	1.511	2.775	10.927
	SIFT	5.140	3.764	9.090	10.287	1.436	1.908	3.226	4.863	**0.531**	1.113	4.819
	D–SIFT	3.175	4.578	5.930	6.811	1.315	1.790	6.302	6.228	0.783	5.169	21.234
Fail Rate	*GASP*	0%	0%	0%	0%	0%	0%	0%	7.14%	0%	0%	0%
	SHOT	7.32%	17.07%	16.67%	25.00%	0%	3.33%	20.69%	39.29%	0%	9.30%	54.29%
	FPFH	1.22%	7.32%	11.11%	27.50%	3.23%	3.33%	17.24%	32.14%	0%	0%	45.71%
	C–SHOT	8.54%	2.44%	12.96%	15.00%	3.23%	3.33%	3.45%	17.86%	0%	4.65%	28.57%
	SIFT	47.56%	58.54%	61.11%	67.50%	3.23%	3.33%	13.79%	21.43%	0%	0%	0%
	D–SIFT	1.22%	0%	1.85%	2.50%	0%	0%	0%	0%	0%	6.98%	34.29%

and 3D connectivity.

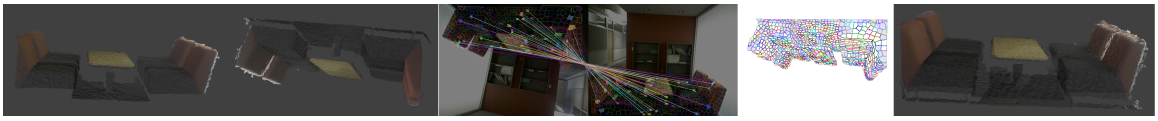
$$f_{geom}(i\,,\,j) = \left(||\overrightarrow{r}||^2 \leq \epsilon_{euclid}^2.l^2\right) \cap \left(|\hat{n_i}.\hat{n_j}| \geq cos\theta_{smooth}\right) \cap \left(\frac{|\hat{l}.\overrightarrow{r}|}{l} \leq \epsilon_{occl}\right) \tag{3.5}$$

where $\overrightarrow{r} = \overrightarrow{p_i} - \overrightarrow{p_j}$, $l = min(||\overrightarrow{p_i}||, ||\overrightarrow{p_j}||)$, and $\hat{l}$=$\begin{cases} \overrightarrow{p_i}/||\overrightarrow{p_i}||\,, & ||\overrightarrow{p_j}|| \geq ||\overrightarrow{p_i}|| \\ \overrightarrow{p_j}/||\overrightarrow{p_j}||\,, & otherwise \end{cases}$.

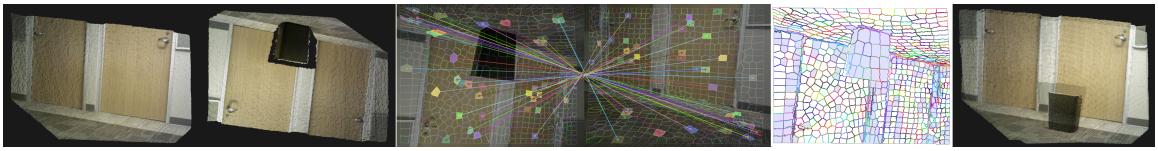
f_{geom} ascertains 3D connectivity through normalized euclidean distance, local smoothness through normal alignment and occluding boundaries through relative change in range. More rigorous, use case dependent constraints such as local planarity and maximum curvature could be enforced too. *Figure* 3.7 illustrates the depth image segmentation with an example. Note that due to surface area regularization, parts of a scene closer to the sensor would have larger superpixels (more pixels) than the ones farther away (*Figure* 3.3). Similarly, a view from farther away would have more superpixels, as it covers more of the scene area (*Figure* 3.8d).

Figure 3.8: **Example results over varied scenes :** These are over different structural settings, and involve varied occlusion, overlap and sensor motion scenarios. Similar presentation and evaluation semantics as in *Figure* 3.1 have been used. Only a sparse sampling of the ascertained associations are indicated in the figures.

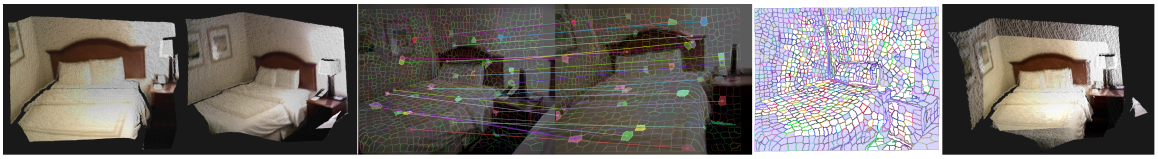
(a) Clutter scene from [63]. It has complex geometry and self-occlusions. The views have significant change in perspective, surface resolution scales.



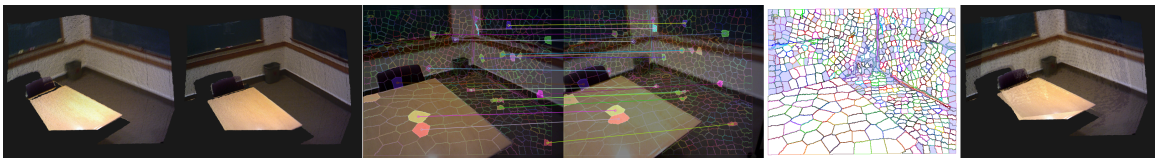
(b) Scene with multiple primitive instances in a near symmetrical setting (points outside the sofa setting volume were clipped off). The views have a full roll inversion, and changes in pitch and yaw as well.



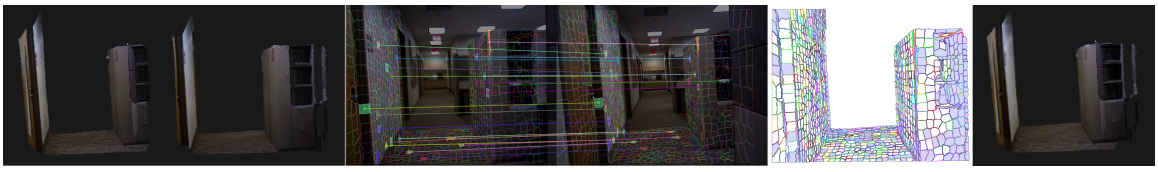
(c) Results over a scene with sparse structure and duplicate primitives. The views have a full roll inversion. An occluding body was introduced in the second view - the superpixels pertaining to it will not get matched (grey overlay).



(d) Results over a scene from *UMD-Hotel* dataset from [56]. The views have significant change in perspective, surface resolution scale, and a partial overlap.



(e) Results over a conference room scene from *Harvard-C11* dataset from [56]. The views have self-occlusion and partially overlapping geometry.



(f) Results over a corridor scene from *Brown-BM1* dataset from [56]

3.4 Experiments And Results

We show results on datasets available from [63, 56], as well as ones collected from everyday scenes - these cover a diverse and challenging range of settings (*Figures* 3.1 and 3.8, *Table* 3.2).

For the experiments in the article, due to the nature of datasets and to maintain uniformity, full neighborhoods were considered throughout ($k_S = |S|$, $k_{S'} = |S'|$). Smaller neighborhoods suffice for settings with locally anisomorphic content though – such as ones with clutter. The superpixel count varied between datasets, and from view to view (due to regularization) - the average number of superpixels per view was around 750. KD-tree queries, C, were at kept at 75. The deviation thresholds, θ_{dev} & r_{dev}, were 10° & $.04m$ respectively.

Exemplar qualitative results are shown in *Figures* 3.1 and 3.8. Nicely localized associations, densely covering the scenes' structures, were achieved. These include associations over regions which have ambiguous or indiscriminate local geometry (and would prove difficult to associate otherwise). The accuracy of associations is also indicated by the quality of unrefined reconstructions resulting from them. As the grey overlays indicate, the occluded and absent parts always get edited out – do not get matched. GASP was able to handle occlusion and partial overlap scenarios, even under sharp sensor motion. Our experiments indicated the associations obtained at low gating values to be accurate. Relatively coarser (but qualitatively correct) associations would arise at higher gating values, with incorrect associations arising mostly at high λ gatings.

Quantitative results are indicated in *Table* 3.2, over kinect datasets from [63] which have ground truths obtained from a motion-capture system. Localization accuracy of GASP's associations was evaluated by utlizing them for ascertaining motion estimates between frames. Translation & Rotation RMS errors and failure rates are shown. Since the datasets had small inter-frame motions, we skipped frames uniformly, starting from regularly spaced initial frames, to simulate significant changes in scene perspectives, sensor baselines and non-overlapping content. The datasets cover different settings - the first two are over varied structural settings with sparse local information, while the last one is over cluttered household/office settings (*Figure* 3.8a).

For automation, roughly chosen high gating values were used ($\lambda \in \{.65,\ ,.65,\ .8\}$, respectively for the three datasets), and the ensuing associations [8] were subsequently filtered for 3D consistency through RANSAC - by simply utilizing the associated superpixels' locations/means as point correspondences. Note that since we ascertain superpixel level associations, better transformation estimates could have been obtained by utilizing richer

[8] A fully dense set of associations was not required for transform estimates. Instead, associations were only ascertained for a volumetrically downsampled set of superpixels, uniformly covering the scenes in 3D

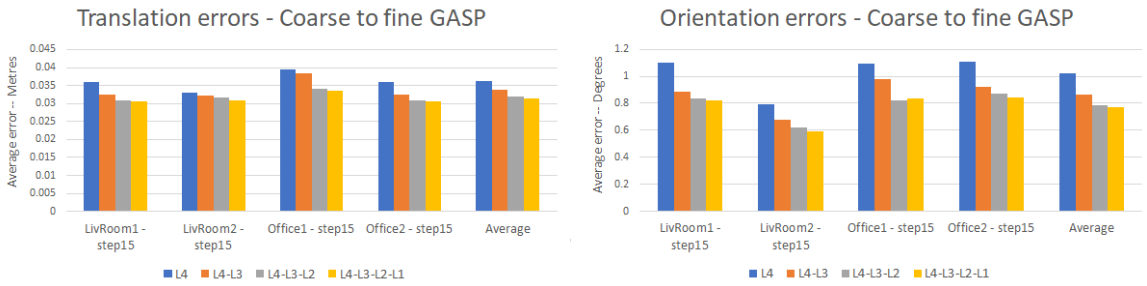


Figure 3.9: **Analyzing coarse to fine association :** Impact of hierarchical association on SE(3) estimation accuracies. $L4$ indicates the coarsest segmentation level, while $L1$ is the finest. A sequence, such as $L3-L2-L1$, indicates hierarchical, coarse to fine GASP starting at the coarsest level $L3$ (in the manner illustrated in *Figure* 3.5). The effect of utilizing increasingly finer segmentation levels for association has been plotted. Translation and orientation errors in the estimate have been indicated on the left and right respectively. The evaluations were done over datasets from [102, 103] - these are indicated on the horizontal axis, with the leftmost ('Average') label in each plot indicating the average over all the datasets. As consecutive frames only had small motion between them, the evaluations were done over pairs 15 frames apart. The analysis indicates that associating hierarchically with increasingly granular patch decompositions results in increased accuracy. It also suggests that the improvements diminish with each additional level.

constraints such as surface patch orientations and overlap - these were not leveraged in experiments.

Promising results were obtained. GASP gave consistently accurate results across the experiments. It was quite robust under increasingly large sensor motions and in varied structural settings.

It also performed favorably, in comparisons with other popular approaches. We compared with geometric as well as appearance based 3D feature approaches (*Table* 3.2, ones below short solid lines), in popular use today. $SHOT$, $FPFH$ ([64, 65]) are point based 3D descriptors based on local geometry; $C-SHOT$ additionally utlizes color information. Dense keypoints (> 2500) for them were evaluated by volumetrically downsampling the point clouds. Standard $SIFT$ ([83]) was utilized, by back-projecting its keypoints in 3D - this is prevalent in RGBD based SfM, SLAM approaches ([56, 55]). For Dense SIFT, keypoints were taken with a step size of 8 pixels, at 3 scales $(1, 3, 9)$. For all methods, the final feature matches were ascertained by filtering for transformation consistency using RANSAC. We tried direct, dense cloud techniques ([54, 77]), but they required short sensor displacements to operate properly. Top values are ordered as rgb.

As can be seen, GASP's results were superior. Its associations gave significantly better motion estimates than geometric feature approaches (whose performance deteriorated with increasing sensor motion). It was also more robust, performed better than popular

appearance based approaches like $SIFT$ and $Dense-SIFT$ under larger viewpoint changes.

3.5 Conclusion

A robust approach was proposed for the problem of dense surface patch / superpixel level data association across views, for pointcloud and range / depth data.

The approach involved an invariant representation of relative geometry over superpixel neighborhoods, and a partial ordering over them - unique to the represented geometry itself. Robust Damerau-Levenshtein edit distance was leveraged for matching these ordered representations. The approach exhibited high robustness to sensor noise and inexact superpixel decomposition across views. It was able to perform in settings with wide baselines, occlusions, partial overlap and steep viewpoint changes. Promising experiment results were achieved in varied and difficult setups.

The approach holds further potential in other applications such as transferring the semantic labels from one view to another, structure and primitive detection, and co-segmentation. These will be explored in the future, as well as improvements in the algorithm by leveraging appearance, and performing experiments in more challenging datasets.

CHAPTER 4

ROBUST GEOMETRIC SCENE ASSOCIATION AND RETRIEVAL

The problems of computing similarity and establishing association between range images and/or 3D point clouds of scenes (observed from a viewpoint, henceforth referred to as *scene-views*) is fundamental to robotics and computational perception in general. It plays an important role in a multitude of applications. Loop closure (identifying a place visited earlier in the trajectory) is intrinsic to metric SLAM ([104]). Localizing with respect to a previously reconstructed map or scene model, and relocalizing after a tracking failure (determining sensor pose without pose priors from trajectory) - both are essential for mapping in practice as well. Different forms of the problem are also key to many navigation scenarios, and in perception tasks such as scene-guided search / foraging or location-based context and activity recognition.

In a minimally restrictive setting, the aforementioned problems (and several others) can be formulated as a retrieval problem - to recognize / identify a scene-view by linking it to stored ones in an assorted, unorganized database. Such a setting would not require any pose priors, spatio-temporal contiguity of collected data[1] or other additional information such as annotations or reconstructed models, and would remove the need to learn a specific pose estimator / regressor for each workspace.

While a lot of progress has been made over the years, including in the retrieval domain, competitive scene association approaches in literature have mostly been reliant on (discriminative) appearance information. Relatively few methodologies work well on noisy, imperfect 3D point clouds or depth images from the real world. Often they critically rely on additional pieces of information available in their target scenario - to prune the association hypothesis space, or obtain strong indirect priors on scene similarity, or enable construction of aggregated spatial information structures to allow its estimation (for instance, [11, 12, 13, 14, 103, 105]) [2]. Approaches also often operate under limited changes in viewpoint and / or on specific types of scene geometry (such as [15, 12, 106]) or they solve a simplified 2D problem (such as [107]). Understandably, methodologies like above are either use case limited or restrictive. Note that approaches like [108] do not ascertain association at all - these directly solve for 3D poses between pre-associated set of data frames.

[1] Spatio-temporally unordered databases can store data acquired from multiple sensors, at multiple times and from disparate locations; could just constitute of snapshots covering scenes of interest.

[2] Quite commonly, approaches rely on spatio-temporal contiguity of frames to to obtain priors or accumulate data structures to ascertain the association.

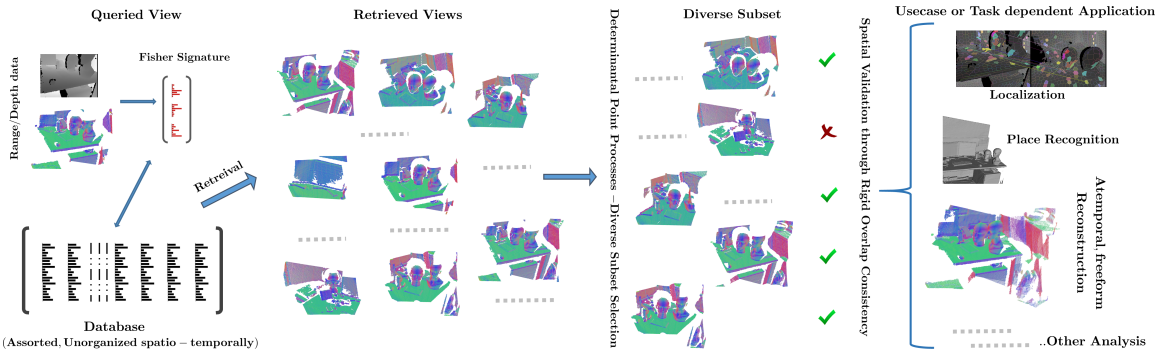


Figure 4.1: **Retrieval pipeline overview :** The query view is indicated in the top-left. Input is a range image or a 3D point cloud. The database (bottom left) constitutes of unordered signatures from arbitrary scene-views, with no labels or ground truth pose annotations. The set of nearest-neighbor retrieved views undergo diversification and subsequent validation. The point clouds are color mapped according to the surface normals - the RGB color of a 3D point is proportional to the component values of its normal.

The dearth of purely 3D geometric scene association approaches in the real world can be primarily attributed to the considerably more ambiguous and challenging depth / range sensing modality. In general, the modality has high local ambiguity and may not be lavish with information on the whole (in contrast to *rgb*). Data acquired from commodity 3D range / depth sensing hardware tends to be particularly noisy as well, has several imperfections. Locally smooth, isomorphic and self-similar nature of typical 3D data from indoor or structural environments makes the problem more difficult. Changes in viewpoint, occlusions and partially overlapping views / content significantly exacerbate the problem further.

We present a minimally restrictive retrieval methodology. Our approach affords means to evaluate *geometric content similarity* between 3D point sets and associate them. We show how it can be utilized to affect *geometric diversity* as well.

We generate descriptive frame-level signatures directly from range images / point clouds (any additional information or assumptions touched upon earlier are not utilized). We make use of macro scale geometry — 3D geometrical interactions (derived from relative angles and distances) over an arbitrarily large span, between arbitrary surfaces, primitives and structures, and their spatial arrangement (for example between walls, floor and furniture, between fixtures and equipment, or just between various parts of a given structural entity). Such interactions when considered collectively are highly discriminative. They are expressed in a learnt viewpoint invariant feature space (4.3). To characterize a scene-view, high order gradient statistics from a dense set of projected interactions are utilized (Fisher Vector, 4.4). To identify a geometrically diverse subset from set of similar retrieved views (4.5), we model a Determinantal Point Process (DPP, 4.6). And to establish association with some of the retrieved views, we employ a fine-grained spatial validation scheme which ascertains consistency of rigid geometry overlap (4.7).

The proposed approach not only outperformed the range / depth data baseline, but was also comparable or better than state-of-art RGB and RGB-D approaches (including ones based on CNN [3]) - and without relying on any additional pose annotations, apriori reconstructed 3D world models, or assumptions such as spatio-temporal contiguity of training data used by other approaches.

Experiments also indicated the learning to be general - unlike most other approaches, it did not require dataset specific training; a single learnt model performed well across the board. Experiments also indicated the performance holding up under significantly sparser databases, and under significantly increased database scale and diversity. Our empirical evaluations quantifying geometric diversity of retrievals were quite encouraging as well. They not only indicated a significant increase in viewpoint diversity of the retrieved set, but also suggested the efficacy of the proposed approach for richer reconstruction and increased workspace coverage - promising hitherto unexplored application scenarios, such as assistive structural search.

4.1 Related work

We refer to only more recent 3D literature among the vast and varied landscape. State-of-the-art loop closure, camera relocalization and place recognition approaches have been primarily based on visual information ([109] presents a recent survey). Many rely on landmark-based features, such as SIFT or ORB, for instance [110, 111, 112, 113]. Approaches such as [114] have focused on the classification problem - one of categorizing similar scenes. [114] utilizes user annotated 3D data to categorize scenes with viewpoint invariance.

Recent state-of-the-art sensor relocalization approaches in real world structural settings [115, 116, 113, 117, 118] are appearance-reliant as well. They also have other critical requirements like scene-specific learning, and / or workspace models or apriori constructed feature clouds (Section 4.9).

As discussed earlier, high-performing scene association approaches operating solely on 3D range/depth data have been relatively scarce. A significant amount of efforts have been put on local 3D point features, such as [64, 65, 15]. There have also been work based on complete point clouds include variants of Iterative Closest Point, Normal Distributions Transform and aggregated 3D features (often position based, such as height above ground, [119]) over densely sampled keypoints. While they work well under some conditions, their performance deteriorates quickly with increasing change in viewpoint and sensor rotations - [1, 120, 121] amongst others, have noted this as well.

[3] Note that our approach considers surface patches as far as half a frame apart from the outset – a distinct difference from popular convolutional network based learning approaches that start by building local features.

[121] matches surface patches between views operating on range / depth data. Our geometric property extraction is along similar lines, and our validation scheme builds upon it. In contrast to [121], which presents a localized surface patch matching algorithm based on aligning geometric sequences defined over neighborhood patches, this work focuses on capturing holistic scene level content for ascertaining geometric content similarity and retrieval.

A number of successful methods exist for shape-based retrieval. [122] presents a recent survey. Shape retrieval approaches are designed to work with CAD object models or clutter free, object-centric data, often with 3D figure-ground information (in contrast to raw, egocentric scene data from noisy sensors) [4]. There have been some successful approaches for 3D object instance detection in clutter, by employing pre-ascertained 3D object templates, for example [123, 124]. More recently, approaches such as [125] have learnt object point clouds to identify 3D shapes with distinct topology.

4.2 Problem Statement

Given a queried scene-view, $\mathcal{V}_Q$, and an extant database, $\mathcal{D}$, of various views from various scenes, $\{\mathcal{V}_s\}_{\mathcal{D}}$ - our algorithm ***a)*** Retrieves a set of views which have structurally similar content as $\mathcal{V}_Q$, ***b)*** Identifies a geometrically diverse subset of views from this retrieved set, and ***c)*** Ascertains whether some of these views pertain to the same scene as $\mathcal{V}_Q$ (*Figure* 4.1).

We denote $\{x_i\}_{i=1}^{c_X}$, $x_i \in X$ to indicate $\mathcal{V}_X$'s segmentation into smooth surface patches. $\{X_h\}_{h=1}^{H}$ denotes the segmentation hierarchy then. Segmentation and hierarchy generation is outlined in *Section* 3.3. To simplify notation, we only indicate the hierarchy level h when it improves clarity.

4.3 Geometric feature space description

Geometric property set extraction : For a given view $\mathcal{V}_X$, at a particular segmentation level - we first express each patch x_i through a 13-D vector set, F'_{x_i} of robust, viewpoint agnostic and macro scale 3D geometric properties. These are derived by utilizing 3D relationships relative to other patches in x_i's neighborhood, $\mathcal{N}_{x_i}$ (along similar lines as [121]). Note that $\mathcal{N}_{x_i}$ is large, non-local - it could span the entire segmentation, $X - x_i$. Neighboring patch count, $|\mathcal{N}_{x_i}|$, is indicated as c_{x_i}.

For a patch $x_i \equiv \mu \in X$, we denote its mean surface normal as $\hat{n}_\mu$ and its 3D mean as l_μ. Denoting α to indicate a patch in μ's neighborhood, with $\hat{n}_\alpha$, l_α denoting its normal and

[4] These also involve specific assumptions - for example, watertight manifolds, surfaces with geometric texture, or disparate / distinctive topology.

mean respectively - an orthonormal basis can be derived from the spanning vectors $\hat{n}_\mu$ and $r_\mu^\alpha = l_\alpha - l_\mu$ through the Gram-Schmidt process. *Figure* 3.2 illustrates this. It also formulates the resultant orthonormal basis, $<\hat{u}_\mu^\alpha, \hat{v}_\mu^\alpha, \hat{w}_\mu^\alpha>$, where $\hat{u}_\mu^\alpha$ is the unit vector in the direction of r_μ^α. Note that coordinate frame spanned by this orthonormal basis is agnostic (invariant) of the sensing viewpoint, since it is a reference frame local to μ & α. Also note that this basis is seldom degenerate, as $\hat{n}_\mu$ and r_μ^α are rarely colinear, especially when data frames are captured from a projective sensing process.

For each neighboring surface patch α in μ's neighborhood, η_μ, we are able to thus extract the following vector of viewpoint invariant properties, $\{f_\mu^{'\alpha}\}_{\forall\alpha\in\eta_\mu}$:

$$
\begin{aligned}
f_\mu^{'\alpha} = [\ & \theta_{\hat{n}_\alpha,\hat{n}_\mu},\ \theta_{\hat{u}_\mu^\alpha,\hat{n}_\mu},\ \theta_{\hat{u}_\mu^\alpha,\hat{n}_\alpha},\ r_\mu^\alpha\cdot\hat{n}_\mu,\ \hat{n}_\alpha\cdot\hat{u}_\mu^\alpha,\ \hat{n}_\alpha\cdot\hat{v}_\mu^\alpha,\ \hat{n}_\alpha\cdot\hat{w}_\mu^\alpha,\ \ldots \\
& r_\mu^\alpha\cdot(\hat{n}_\alpha\times\hat{n}_\mu),\ ||r_\mu^\alpha||, ||r_\mu^\alpha||\cdot sgn_{\epsilon_\theta}(\hat{n}_\mu\cdot\hat{u}_\mu^\alpha),\ ||r_\mu^\alpha||\cdot sgn_{\epsilon_\theta}(\hat{n}_\alpha\cdot\hat{u}_\mu^\alpha),\ \ldots \\
& ||r_\mu^\alpha||\cdot sgn_{\epsilon_\theta}(\hat{n}_\alpha\cdot\hat{v}_\mu^\alpha),\ ||r_\mu^\alpha||\cdot sgn_{\epsilon_\theta}(\hat{n}_\alpha\cdot\hat{w}_\mu^\alpha),\ ]^T
\end{aligned}
\tag{4.1}
$$

The θ above refers to the angle between the indicated vectors and $\times$ represents an outer product. $sgn_e(..)$ is a robust signum function that clamps to zero when its parameter $\notin [cos^{-1}(PI - e_\theta),\ cos^{-1}(e_\theta)]$, with e_θ accounting for allowable tolerance to angular noise.

The feature vector $f_\mu^{'\alpha}$ basically represents an overcomplete characterization of relative properties between the two patches - formulated in a viewpoint agnostic fashion. The first part (first 9 features) captures angular relationships between r_α^μ, $\hat{n}_\alpha$ & $\hat{n}_\mu$, characterizes $\hat{n}_\alpha$ in the invariant frame derived from r_μ^α and $\hat{n}_\mu$, and characterizes r_α^μ. The second part (remaining 4) consists of robustified features - as a measure against noises arising due to estimation from real world, noisy data. Signs of projected normals' components are captured through robust signum functions and are augmented with the magnitude of relative displacement vector.

Feature space projection : A patch's property set $F_{x_i}^{'} = \{f_{x_i}^{'\alpha}\}_{\alpha\in\eta_{x_i}}$ is then projected onto a subspace learnt beforehand [5]. The projection reduces redundancy in $f_{x_i}^{'\alpha}$, makes its components more independent, and further stabilizes it in face of noises. Importantly, this alleviates the burden on subsequent learning, and fits with the component independence assumption made in *Section 4.4* to train Gaussian mixture models with diagonal covariances. A 12-D subspace was learnt through cross validation experiments minimizing for data reconstruction error. To learn the subspace, we used an asymptotically efficient version of independent component analysis, [126], that optimizes adaptively chosen nonlinearities.

The feature space projection results in a 12-D feature vector set $F_{x_i}=\{f_{x_i}^\alpha\}_{\alpha\in\eta_{x_i}}$. By considering the patches in x_i's macro scale neighborhood, η_{x_i}, the feature set F_{x_i} can thus robustly

[5] Our empirical evaluations indicated this learnt subspace to be general. *Figure* 4.3 validates our observations.

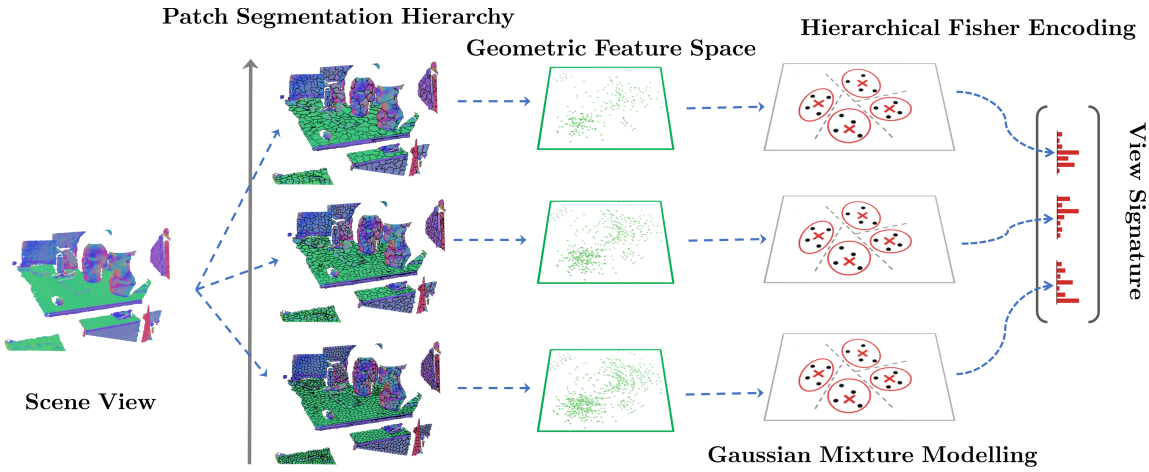


Figure 4.2: **View signature encoding :** Geometric properties are extracted over a hierarchy of patch segmentations. At each segmentation level, the aggregate sets of properties is first mapped to a viewpoint invariant geometric feature space, to get a decorrelated, dimensionally independent principal feature set. These are then jointly encoded as a view level signature using fisher vector embedding.

express the 3D geometry in x_i's non-local neighborhood. An aggregation of such feature sets arising from all the patches, $F^X = \{F_{x_i}\}_{i=1}^{c_X} \equiv \{f_{x_i}^{\alpha} | x_i \in X, \alpha \in \mathcal{N}_{x_i}\}$, can thus invariantly and richly express the geometry of the entire scene as captured by $\mathcal{V}_X$. Finally, the above procedure is repeated for each level in the segmentation hierarchy, to capture fine as well as coarse details. This results in a hierarchy of aggregate feature vector sets, $\{F_h^X\}_{h=1}^{H}$.

4.4 Encoding feature space statistics

To obtain a descriptive signature for a given view, $\mathcal{V}_X$, we encode the aggregated feature sets using Fisher vector embedding (FV, [127, 128]) - this captures the normalized gradient of the log-likelihood of the feature sets. The Fisher kernel theory, first presented in [127], introduces a similarity kernel, arising as a consequence of maximizing the log-likelihood of generatively modeled data. In this paper, Gaussian Mixture Models (GMM) were used to model the feature space distribution.

Given a learnt GMM, P_Θ, parameterized as $\Theta = \{p_g, \nu_g, \Lambda_g\}_1^G$, the FV embedding of the aggregate feature set F^X, indicated as $\phi(F^X)$, is obtained as $\phi(F^X) = L_\Theta \nabla_\Theta log(P_\Theta(F^X))$. Here, L_Θ is the Cholesky decomposition factor of the inverse Fisher Information Matrix, and $\nabla_\Theta log(P_\Theta(F^X))$ is the score function (log-likelihood gradient). Following similar analysis as [128], under assumptions of diagonal covariance matrices, Λ_g, and independence of the samples, $f_{x_i}^{\alpha}$, the

embedding evaluates as

$$\phi(F^X) = \left[m_1^0,\, m_1^{1^T},\, m_1^{2^T} \ldots m_g^0,\, m_g^{1^T},\, m_g^{2^T} \ldots m_G^0,\, m_G^{1^T},\, m_G^{2^T}\right]^T \tag{4.2}$$

where m_g^0, m_g^1, m_g^2 respectively capture the normalized zeroth, first and second order statistics of the sample set that falls in the g-th mixture component of the GMM. $\phi(F^X)$ has a dimensionality of $d_\phi = (2d_F + 1) \cdot G$, where G is number of mixture components, and $d_F = 12$ is the dimensionality of our geometric feature space. Below, $\mathbf{1}$ denotes an all-one vector and $\pi_{ij,g} = \frac{\exp\left[-\frac{1}{2}(f_{x_i}^{x_j} - \nu_g)^T \Lambda_g^{-1} (f_{x_i}^{x_j} - \nu_g)\right]}{\sum_{g=1}^{G} \exp\left[-\frac{1}{2}(f_{x_i}^{x_j} - \nu_g)^T \Lambda_g^{-1} (f_{x_i}^{x_j} - \nu_g)\right]}$.

$$m_g^0 = \frac{1}{c_X c_{x_i} \sqrt{p_g}} \sum_{i=1}^{c_X} \sum_{j=1}^{c_{x_i}} (\pi_{ij,g} - p_g) \tag{4.3a}$$

$$m_g^1 = \frac{1}{c_X c_{x_i} \sqrt{p_g}} \sum_{i=1}^{c_X} \sum_{j=1}^{c_{x_i}} \pi_{ij,g} \Lambda^{-1/2} (f_{x_i}^{x_j} - \nu_g) \tag{4.3b}$$

$$m_g^2 = \frac{1}{c_X c_{x_i} \sqrt{2p_g}} \sum_{i=1}^{c_X} \sum_{j=1}^{c_{x_i}} \pi_{ij,g} \left[\Lambda^{-1} (f_{x_i}^{x_j} - \nu_g)(f_{x_i}^{x_j} - \nu_g)^T - \mathbf{I}\right] \mathbf{1} \tag{4.3c}$$

$\phi(F^X)$ is then component-wise square root normalized (by replacing each component, 'a' of $\phi(F^X)$ by '$|a|^{1/2} sign(a)$'), and ℓ_2 normalized. The square root normalization serves to alleviate the dominant effect of relatively indiscriminate samples occurring with high frequency (for example, arising from patches on a wall or ceiling) and the ℓ_2 normalization helps generalization across different scenes by normalizing the energy content. The desired view signature vector for $\mathcal{V}_X$, denoted as $\psi(X)$, is obtained by evaluating the embedding at each level in hierarchy, and concatenating them. We have then —

$$\psi(X) = \left[\phi(F_1^X)^T, \ldots\, \phi(F_h^X)^T, \ldots, \phi(F_H^X)^T\right]^T \tag{4.4}$$

4.5 Similarity and Retrieval

The thus obtained view signature, $\psi(X)$, captures discriminative 3D geometrical properties, and is robust to viewpoint changes, sensor noise, occlusions and other data imperfections by design. As experiments indicate, a metric based on such view signatures is a reliable measure of 3D geometric similarity. We tried ℓ_1 & ℓ_2 distance metrics, and used ℓ_1 for all experiments in the paper as it performed better. Thus the similarity between two given views V_X & V_Y can be denoted as, $\mathbf{s}(X, Y) = -(\sum_1^{25GH} |\psi(X) - \psi(Y)|_1)$.

Given a a queried view, $\mathcal{V}_Q$, and a database $\mathcal{D}$ of view signatures, one can thus retrieve a set of putative view associations in the geometric sense through nearest neighbor queries. We

indicate this retrieved set of putatively associated views as $\mathcal{R} = \{\mathcal{V}_X\}_{X=1}^{c_\mathcal{R}}$.

4.6 Diversity Sampling with Determinantal Point Processes

Depending on the distribution of scenes' views in the database, $\mathcal{R} = \{\mathcal{V}_X\}_{X=1}^{c_\mathcal{R}}$ could be overwhelmed with views which are *near duplicates* (all being very similar to each other, hence almost equally similar to the queried view). This may not be desirable since the subset of top retrievals could just be flooded with near duplicates of false putative associations, resulting in complete failure. By filtering out near duplicates, a diversity based subset selection procedure may still be able to salvage correct, albeit lower ranked, putative associations present in $\mathcal{R}$ with further post-processing validation.

A diverse set of retrievals is generally desirable. It would provide assorted and possibly complementary information, which could be made use of thereon. For instance, it could be potentially beneficial in reconstruction or coverage tasks, where diverse viewpoints observing the environment with only partially overlapping content are more desirable than having redundant views from nearly the same perspective. A querying human user could also be better assisted by being provided with a diverse set of the retrievals to choose from.

Determinantal point processes ([129]) are employed to select a diverse subset of candidate views, $\mathcal{C}$, from $\mathcal{R}$. A point process $\mathcal{P}_L$ is called an L - ensemble *k-determinantal point process* if for every random subset, $\mathcal{C}$, of $\mathcal{R}$, such that $|\mathcal{C}| = k$, drawn according to $\mathcal{P}_L$, we have $\mathcal{P}_L\left(\mathcal{C};\mathcal{R}\right) = \frac{det(L_\mathcal{C})}{\Sigma_{\forall \mathcal{A} \in \mathcal{R},\, |\mathcal{A}|=k}\, det(L_\mathcal{A})}$. L here is a symmetric positive semi-definite similarity matrix indexed by the elements of $\mathcal{R}$. $L_\mathcal{C}$ is the principal minor (submatrix) with rows and columns from L indexed by the elements in subset $\mathcal{C}$. Thus the probability of selecting a subset $\mathcal{C}$, ($|\mathcal{C}| = k = c_\mathcal{C}$) elements is directly proportional to the determinant of the submatrix indexed by it. Note that higher diagonal values would proportionately encourage their inclusion in a selected subset $\mathcal{C}$ as they lead to higher determinants. Similarly, the off-diagonal values determine correlation between different elements, and a high value decreases the determinant overall. Thus two elements with a high similarity value tend not to co-occur in $\mathcal{C}$. DPP sample sets are therefore able to balance the net significance of their constituent elements with their diversity. We modeled L accordingly as follows

$$\{L\}_{X,Y} = \rho_X \rho_Y \kappa e^{\frac{\mathbf{s}(X,Y)}{\sigma}}\ ,\ 1 \leq X, Y \leq c_\mathcal{R} \tag{4.5}$$

where $\rho_X = e^{\frac{1}{2}\frac{\mathbf{s}(X,Q)}{\omega}}$, $\exists X \in \mathcal{R}$ models the similarity of a retrieved view $\mathcal{V}_X$ to the queried view $\mathcal{V}_Q$. The similarity between two given views $\mathcal{V}_X$ and $\mathcal{V}_Y$ is captured by the rightmost term. Positive valued parameters σ, ω and κ can be tuned to balance the need for both diversity and similarity to $\mathcal{V}_Q$. A lower sigma would induce a higher resolution in similarity scores between retrieved views, and hence would result in a more diverse subset selection.

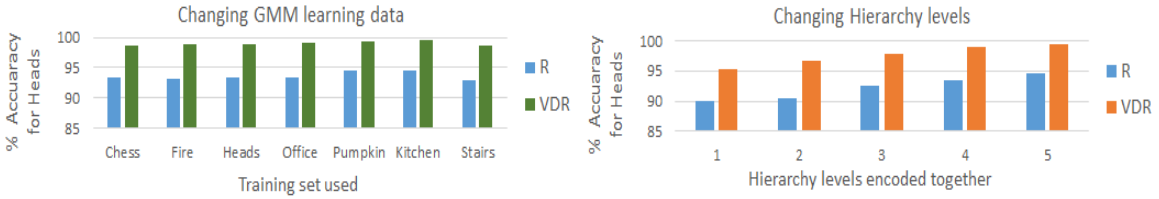


Figure 4.3: ***Left* - Consistency in GMM learning :** Similar retrieval accuracies were achieved with GMMs learnt from each of the 7 training sets. ***Right* - The impact of encoding a fine to coarse hierarchy of levels :** As can be seen, significant improvements are achieved when properties are captured at multiple scales.

While the *MAP* inference on $\mathscr{P}_L$ to determine the most probable subset is NP-hard, efficient sampling algorithms exist which provide good approximate solutions in practice. For our purposes, a greedy procedure based on [129] which results in $\mathcal{O}(k \log k)$-approximation worked well.

4.7 Validating candidate views for association

We employ a finer grained spatial validation step before finally associating the queried view with some of the views in the candidate set, $\mathcal{C} = \{\mathcal{V}_X\}_{X=1}^{c_\mathcal{C}}$. This is done by directly leveraging the rigid 3D spatial arrangement of surface patches to ascertain surface alignment. We make use of the patch matching scheme presented in our prior work [121]. It utilizes a sequence alignment scheme over similarly motivated patch properties to find standalone correspondences based on 3D neighborhood similarity. A semi-dense set of correspondences can be ascertained. Rigid transform between two views of a given scene can then be robustly, accurately computed through consensus of patch associations.

When views from scenes with different geometrical content are matched through [121], the matches would likely be inconsistent with respect to the computed transform. We exploit this understanding to validate associations with candidate views. For each candidate view, $\mathcal{V}_X \in \mathcal{C}$, and the queried view, $\mathcal{V}_Q$, we utilize randomly sampled patches to estimate rigid transformations both ways, that is, $T_Q^X \equiv (R_Q^X, t_Q^X)$ and $T_X^Q \equiv (R_X^Q, t_X^Q)$ and check whether they are consistent with each other. We ascertain a candidate $\mathcal{V}_X \in \mathcal{C}$ as associated with $\mathcal{V}_Q$ when $\left\|\log\left(R_Q^X R_X^Q\right)\right\|_2 \leq \epsilon_{val}^\theta$ & $\left\|t_Q^X + t_X^Q\right\|_2 \leq \epsilon_{val}^\varepsilon$ - we are basically ensuring that the magnitude of the rotation and translation components in the residual transform, $T_Q^X T_X^Q$, are below certain thresholds $\{\epsilon_{val}^\theta, \epsilon_{val}^\varepsilon\}$.

Table 4.1: **Quantitative evaluations and comparisons :** The presented approaches (*R* , *VDR*) are compared with baselines through localization accuracies on the standard 7-scenes datasets from [130, 131]. All methods utilize RGB-D data during training, except [131] D, and our *R* and *VDR*, which are based on range / depth data. During test time, the three leftmost approaches only take RGB images as input, while the three rightmost approaches only take range / depth images - the rest operate on RGB-D. *Average* indicates the average among the 7 datasets. *Combine* indicates performance when jointly considering all 7 scenes as a single database. VDR outperforms all the RGB-D approaches while using depth information *only*. R performs very well as well, outperforming all but two RGB-D approaches.

Data	**Appearance Reliant** (RGB or RGB-D)							**Depth – Only**		
Approach	**Reconstruction Truth Needed for Relocalization**								**Retrieval**	
Method	*Spr*[131]	[132]*C*	*DSc*[116]	[131]	[133]	[115]	[132]	*D[131]*	*R*	*VDR*
Chess	70.7	94.9	97.4	92.6	96	99.4	**99.6**	82.7	97.3	99.5
Fire	49.9	73.5	74.3	82.9	90	94.6	94.0	44.7	92.3	**97.8**
Heads	67.6	48.1	71.7	49.4	56	95.9	89.3	27.0	93.5	**98.9**
Office	36.6	53.2	71.2	74.9	92	97.0	93.4	65.5	89.7	**98.4**
Pumpkin	21.3	54.5	53.6	73.7	80	**85.1**	77.6	15.1	78.3	82.8
Kitchen	29.8	42.2	51.2	71.8	86	89.3	91.1	61.3	87.9	**93.7**
Stairs	9.2	20.1	4.5	27.8	55	63.4	**71.7**	13.6	54.8	61.0
Average	40.7	55.2	60.1	67.6	79.3	89.2	88.1	44.3	84.8	**90.3**
Combine	38.6	55.2	62.5	-	-	-	-	-	84.8	**90.4**

4.8 Further details and discussion

The approach is amenable to any boundary-preserving patch segmentation scheme, as long as it results in superpixels / patches that are geometrically regularized for smoothness and compactness. For example [100], which segments volumetrically, could be used while working with point clouds; and surface segmentation schemes such as one presented in *Section* 3.3 ([121]) could be employed when working with depth / range images. Both [121] and [100] performed well in our experiments. *Section* 3.3 also discusses agglomerative and divisive methodologies for generating a segmentation hierarchy. We used four levels of segmentation hierarchy ($H = 4$). The number of mixture components were also kept fixed, $G = 1250$. The GMMs were learnt through an expectation maximization scheme, and the mixture components were initialized from the result an iteration of K-Means++ procedure. Our empirical analysis indicated the learnt feature space distribution to be general for similar sensor types [6]. *Figure* 4.3 suggests that as well. In fact, a single set of Gaussian mixture (and ICA) models were utilized for all the experiments shown in the article (except *Figure* 4.3).

[6] Sufficient number of GMM components should be utilized to span the extent of the geometric feature space. This is a function of maximum scene scale captured, and thus sensor range.

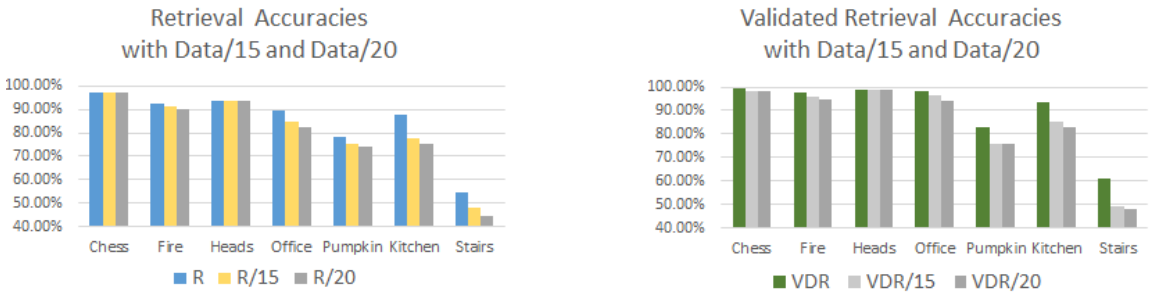


Figure 4.4: **Accuracies with significantly sparser acquisition :** Database sizes were reduced to $1/15$ and $1/20$.

In practice, for efficiency, while encoding feature space statistics (Section 4.4), it suffices to approximately ascertain F_h^X by sampling patches from X_h, and subsequently sampling the neighborhoods of the sampled patches. This also partly corroborates our assertion that the methodology is robust to occlusions. Databases were indexed as KD-trees. Our current straight up implementation is not optimized for efficiency (on a 4.2 GHz, 4 core setup, 4.3 - 4.6 takes $\sim .3$ ms, 1000 superpixels), though the methodology is GPU parallelizable. Most of the procedures outlined in Sections 4.3, 4.4, 4.5, 4.6 and 4.7 can be GPU paralellized in a straightforward fashion. The computational bottleneck arises during validation, which is quadratic in number of superpixels ($\sim$ 1s for segmentation with 1000 superpixels at finest level, but again naturally parallelizable). Note that it suffices to validate at a coarse hierarchical level ($\sim$ 250 superpixels) — the result, T_Q^X, can then be used as reliable initialization and be quickly refined iteratively as per task.

4.9 Experiments

In all experiments, the method indicated '*R*' refers to our retrieval approach (till Section 4.4), without the diverse subset selection and validation steps. '*DR*' refers to our approach till Section 4.6, with diversification but without the validation step. '*VDR*' would then refer to the complete approach, resulting in the set $\mathcal{C}_{vld}$ - diverse retrievals which have been validated through rigid overlap consistency. The retrievals in both the sets $\mathcal{C}$ and $\mathcal{C}_{vld}$ follow the same order (by $\mathbf{s}(X, Y)$) as they appear in the initial retrieval set $\mathcal{R}$. All analysis is done on the top few results from these sets.

The retrieval and association problems can be subjective - two views with only partially overlapping geometric content can be evaluated differently by users. We employed an objective measure - evaluating our retrieval approach on a sensor relocalization task. We utilized the 7-scenes datasets from [130, 131], the standard benchmark for indoor RGB/RGB-D relocalization. The objective is to localize the sensor (ascertain pose) with respect to the workspace within the maximal allowable translation and orientation errors (5 cm and 5 deg

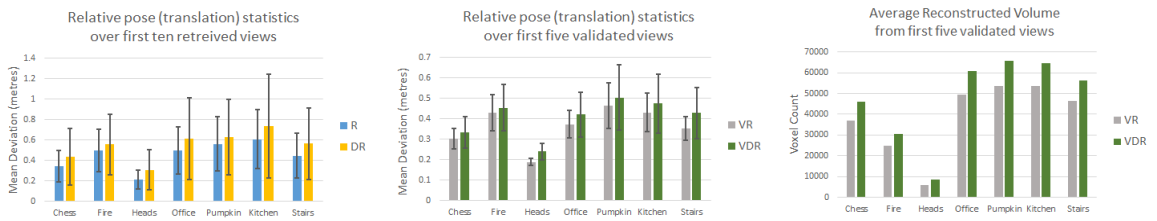


Figure 4.5: **Quantifying diversity :** Left, Middle: The average relative translation of the retrieved views with respect to the queried view. One can see DR improves diversity over R, and VDR improves over VR. Right: Efficacy of diverse viewpoints for reconstruction task. The average number of voxels (in a 8 cm^3 occupancy grid) occupied by ground truth reconstructs from the first five validated retrievals from VR and VDR are plotted. From the same number of initial views, VDR results in richer reconstructs that capture significantly more voxels in the scene.

respectively). The datasets are collected from different workspaces (although some scenes in *Redkitchen* and *Pumpkin* are quite similar). Standard train - test splits are provided, with the viewpoints in the test set differing significantly from the training set. This makes it most appropriate for use in the evaluation [7]. 7-scenes also provide additional training information - global sensor pose annotations, as well as reconstructed volumetric workspace models.

In our approach, *R, DR and VDR*, depth images for training were simply encoded as an unordered view-signature database. A given query image from the test split was localized by computing the relative transform with respect to the top retrieval (in the sets $\mathcal{R}$, $\mathcal{C}$ and $\mathcal{C}_{vld}$ respectively), and the localization accuracy was computed by evaluating the disparity between the estimated and ground truth relative poses. Same as in the baselines, 5 cm and 5 deg are the allowable error. Our approach did not require additional information accompanying the datasets to operate (pose truth annotations and workspace reconstructs). Importantly, it also did not require specific training for each dataset. This differs from most of our baselines which required some additional information or dataset-specific training.

***Baselines*:** We compare our approach against many baselines. Approaches like [18, 136, 133, 118, 116, 117, 137] require additional information and dataset specific training. They rely on annotations, workspace models, and involve regression against absolute sensor poses or 3D coordinates of pixels. Deep-CNN based regressors have been proposed as well, such as [118, 116, 117, 137]. Such approaches can overfit on the training data, and are difficult to generalize to scenes that are not similar to the training. Some baseline results were not shown in *Table* 4.1 — [130], which presents a random ferns based retrieval method over RGB-D, report accuracies differently; but they indicate the achieved results to be weaker than some of the baselines considered in *Table* 4.1. Methods like [118, 117, 137] report localization accuracies as median errors - since the lowest reported median errors, that we are

[7] As opposed to mapping, visual odometry or semantic scene datasets such as [63, 102, 56, 134, 135]. These either do not have enough loop closures and/or are synthetic, or lack ground truth for quantitative evaluation or standard train-test splits for loop closure.

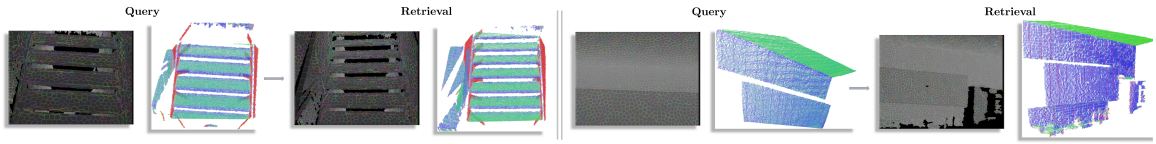


Figure 4.6: **Failure cases :** Two possible failure (or problematic) scenarios are indicated. 3D normal maps of queried views are shown in the top row along with the original images. The retrieved views are shown under them respectively. Note that although the retrievals are correct - that is, they have the same structural content as the queries - the subsequent validation, or inaccurate localization failed them in final evaluations. This is because both the scenes are geometrically ambiguous. The left scene is not discriminative enough from geometry alone, which results in inconsistent transform estimates and is hence not validated. The one on the right has strong geometrical aliasing - while it does get validated, the transformation estimates / localization is erroneous.

aware of, are greater than 10 cm (translation, implicitly includes orientation errors as well), these methods are also not as accurate as some of the baselines in *Table* 4.1. Approaches [113] and *Sparse* [131] employ frame to model matching for relocalization. They match local features from the query frame to a global feature cloud accumulated and reconstructed *a priori* from the training data and the pose ground-truth annotations. [113] shows nice results, though we were unable to obtain exact numbers from the authors. However, *VDR* in *Table* 4.1 does seem to perform better than [113] in 4 out of 7 datasets in comparison. *VDR* also seems to outperform [113] in at least 6 out of 7 datasets when only 1/15 of the training data is used (*Figure* 4.4). All the aforementioned approaches are appearance-reliant as well (except [131] which additionaly present a depth only variant). We also tried a retrieval methodology similar to ours with local 3D geometric point-features (such as [64]), but their performance was worse than those shown in *Table* 4.1.

As *Table* 4.1 shows, *VDR* achieved state-of-the-art results through pure geometry alone - without needing any additional annotations, assumptions or appearance features. Equally promising were the results from *R* which were obtained by simply using the first retrieval in $\mathcal{R}$ (no diversification or validation), which were better than all baselines but two. *DR* gives the same results as *R* in the relocalization experiments and is hence not shown. This is because the accuracies were evaluated with respect to only the top retrieval - this is the same for *R* and *DR* since the greedy algorithm we used for k-DPP automatically selects the top-scoring retrieval as the first one. These results support our hypothesis that macro-level 3D geometry holds immense discriminative information.

In the last row of *Table* 4.1, we combined all training data from the 7 datasets into one single database, and evaluated accuracies of the combined test splits. As can be seen, the results held up quite well in the combined experiment, when the database size and complexity (variety, aliasing) was drastically increased.

We also tabulated the affect of significantly reducing the database sizes - by re-evaluating results with databases built from only $1/15^{th}$ and $1/20^{th}$ of the available train-splits for each dataset. With a much sparser coverage of the environment, both retrieval and subsequent validation and localization becomes much more difficult. The frames were sampled at uniform intervals, thus may have steep viewpoint changes, much reduced content overlap and significantly increased occlusions. As *Figure* 4.4 indicates, the accuracies of both *R* and *VDR* held up quite well. This is indicative of the approach's robustness to these practical challenges.

The approach generalizes well. Our experiments do not suggest a need for scene specific training - a single set of learnt gaussian mixtures and ICA projection matrices were utilized in all our experiments (except *Figure* 4.3). The training data was taken from the train split of Redkitchen in [131], and from datasets in [56, 63], a reasonably rich and diverse set of samples. *Figure* 4.3 shows the robustness of the GMM parameters with respect to the dataset used to train it. As can be seen the results stay consistent.

Finally, we conducted experiments to quantify the effect of our diversification approach, and its role in generating significantly richer reconstructions. As *Figure* 4.5(left, middle) show, the diversity of retrieved viewpoints is greatly improved due to our DPP-based diversification. Note that DR and VDR select views which are not only further off than the queries (higher relative mean), but result in view sets which have significantly more viewpoint variance amongst themselves as well (significantly higher standard deviations). And as *Figure* 4.5(right) shows, the reconstruction volume improves significantly when a diversified set of views is utilized. *Figure* 4.7 shows a qualitative example. One can see that the diversified retrievals are significantly more diverse, from varied viewpoints, and are resulting in an appreciably richer reconstruction. In general, retrieval as well as geometric diversity is often desirable - apart from reconstruction, it would prove useful in other tasks such as structure and semantic analysis, and 'human in loop' selection tasks.

4.10 Conclusion

We presented a robust solution to the problems of measuring 3D geometric similarity between 3D range images or point clouds, and determining whether they come from the same scene. A general-purpose retrieval approach was proposed, based on encoding (FV) of viewpoint-invariant features that are hand-crafted to capture 3D geometry at macro scales. The approach performed well in real world settings - including ones that involved sharp viewpoint changes, partially overlapping and occluded content. It scaled well, and did not require scene-specific training - making it useful in a variety of scenarios. As experiments established, the approach is powerful and did better than specifically fitted solutions such as CNNs trained on RGB or RGB-D data. Furthermore, we introduced a way to obtain

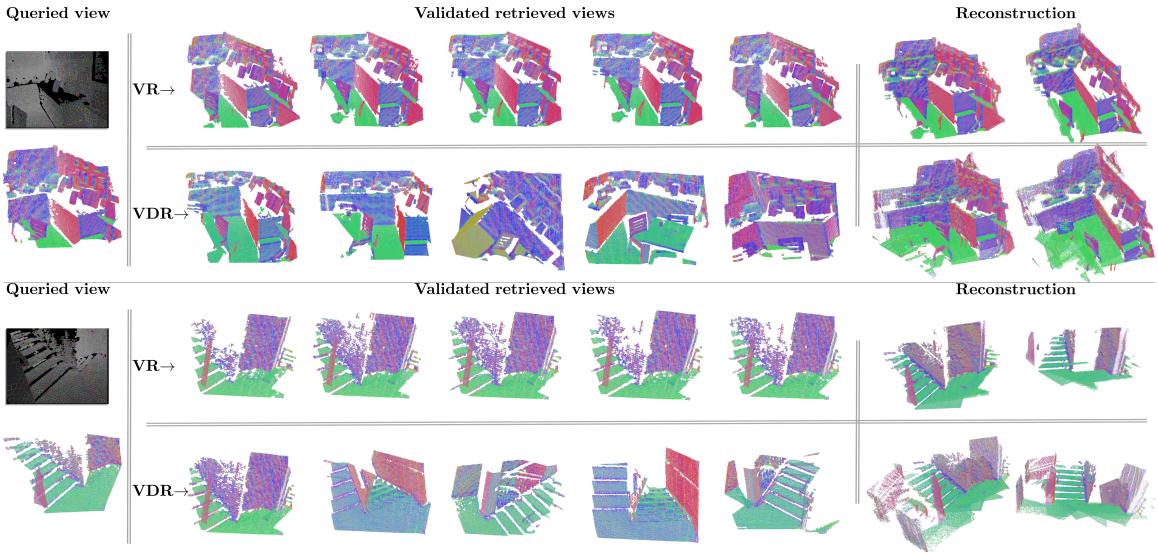


Figure 4.7: **Example scene retrieval and reconstructions :** For each scene, the top row shows retrievals from VR and the bottom row shows retrievals from VDR. Queried view is shown on the left as a depth image with overlayed patch boundaries. Views on top row are the top-five retrieved and validated views without using DPP. Views on the bottom row are the top-five validated views with DPP. Reconstructed scene models from the respective sets are shown on the right from two perspectives. Note that viewpoints vary significantly in the diversified retrievals, and results in a much larger reconstructed volumes (over 1.5x).

geometrically diverse retrievals (DPP), and showed how such retrievals can help generate richer reconstructions. Interestingly, in contrast to CNN approaches which begin with a local neighborhood, our approach utilized macro scale features from start. The combination of both paradigms would be explored in future work, for this and other tasks involving 3D recognition.

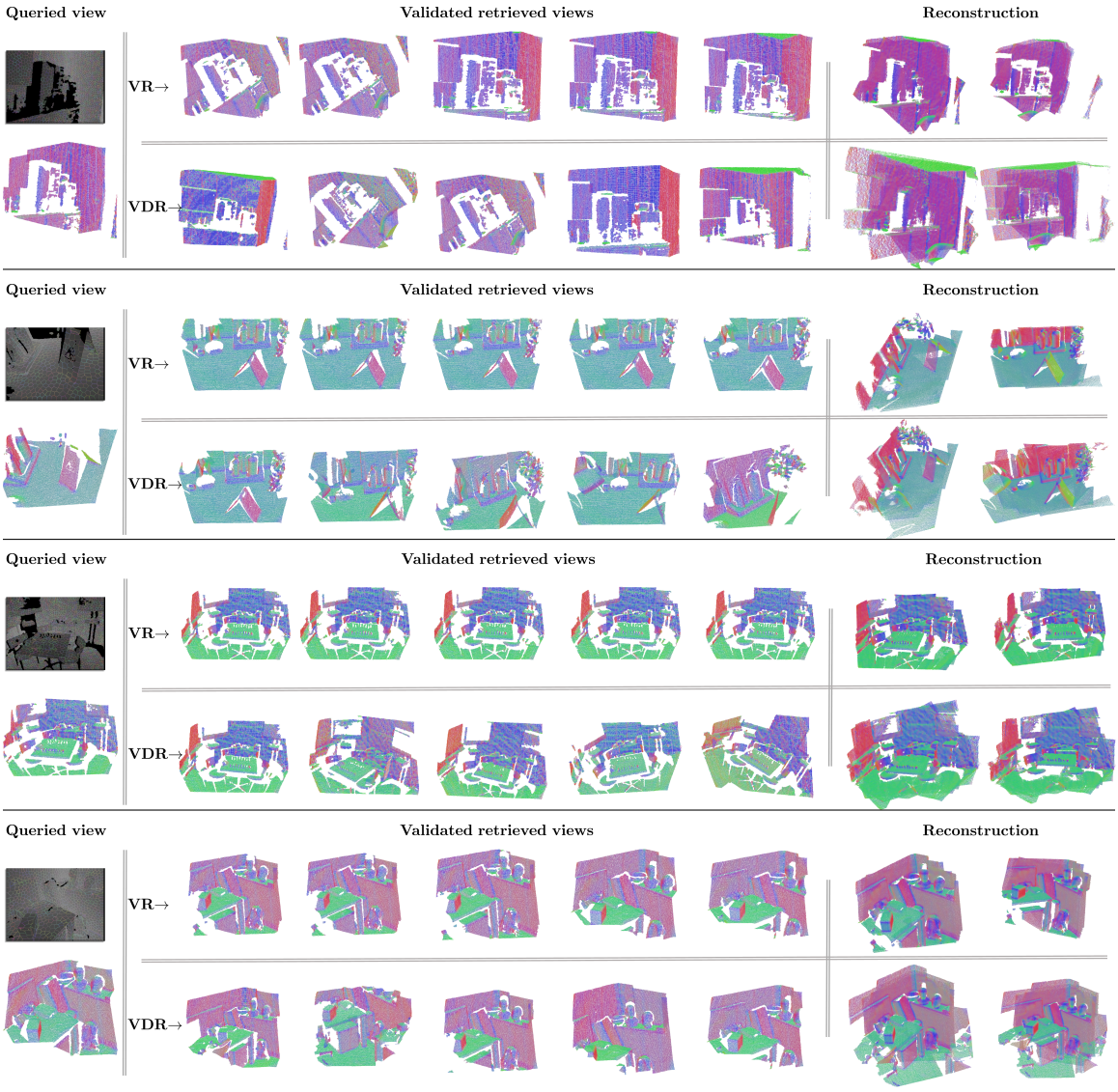


Figure 4.7: **(Continued) Example scene retrieval and reconstructions :** Example scene retrieval and reconstructions are shown. For each scene, the top row shows retrievals from VR and the bottom row shows retrievals from VDR. Queried view is shown on the left as a depth image with overlayed patch boundaries. Views on top row are the top-five retrieved and validated views without using DPP. Views on the bottom row are the top-five validated views with DPP. Reconstructed scene models from the respective sets are shown on the right from two perspectives. Note that viewpoints vary significantly in the diversified retrievals, and results in a much larger reconstructed volumes (over 1.5x).

CHAPTER 5

A NONSMOOTH NONCONVEX LOSS AND RELATED ROBUST OPTIMIZATION

In *Chapters* 3 and 4, we discussed how robustness can be achieved by appropriate representations and methodologies designed expressly for real world 3D data. Dedicated approaches were presented, tackling the specific challenges posed by 3D modality in the real world and focused on the problems at hand. The results obtained were promising, achieving a high degree of robustness in difficult settings and scenarios.

Robustness can also be approached more intrinsically, in a fundamental sense. It can be achieved by addressing the optimization involved in estimation of statistics of interest (desired numerical quantities - for instance, estimating SE3 transforms from noisy 3D correspondence data). This is the focus of this chapter. Such an approach would be abstracted from its application specifics. Apart from being quite effective by itself, it would complement the application-centric (front-end) methods, such as ones we have been discussing in *Chapters* 3 and 4.

By being tolerant to outliers while estimating a statistic, robust optimization can allow us to approach robustness in a direct fashion. As we will see, it enables us to essentially reject the influence of gross outliers, and even regulate the relative bias of inlying samples on the estimate.

We discuss robust loss functions and optimization in this chapter, and present some useful, novel contributions.

A robust loss function generally applicable to estimation and learning problems is proposed. It has a desirable combination of properties well suited for exact estimation and outlier suppression / rejection. The loss function is nonconvex, nonsmooth, and strictly concave in $[0, \infty)$. Desirably, it allows data with zero or near zero residues (the best inliers), to have the maximum and similar influence, while large residues and outliers are aggressively suppressed. Its nonsmooth nature results in estimates that exactly fit more inliers. In our experiments so far, on 3D data, it has performed promisingly - it seemed well suited for estimation and fitting tasks, and compared well with loss functions in popular use.

$$\rho_{\times}(\boldsymbol{r}) = \frac{e^{2|\boldsymbol{r}|} - 1}{e^{2|\boldsymbol{r}|} + 1} \tag{5.1}$$

Equation 5.1 above formulates the proposed loss, $\rho_{\times} : \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$. Here $\boldsymbol{r} \equiv \boldsymbol{r}(\theta) \in \mathbb{R}^{d(\boldsymbol{r})}$ is

the residue vector corresponding to a datum. It is itself a function of the parameter vector, θ - which parameterizes the model, configuration or state associated with the datum.

Nonconvex, nonsmooth loss / risk based formulations have been known to outperform their convex and / or smooth counterparts in a variety of problem settings. They are the correct choice when the data has non-Gaussian noise, missing features, irregular and / or multiple structures. Besides, real world problems and phenomena often have a genuinely nonconvex, nonsmooth description.

On the flipside, such objectives are difficult to optimize. This is due to lack of differentiability (especially at the minimizer), presence of saddle points [1], and related numerical intricacies.

Thus in conjunction, a framework to optimize a related class of nonsmooth, nonconvex loss functions is proposed. Approach and analysis for robust optimization of some important, popular objective forms is provided. Sufficiency conditions for global convergence are discussed as well. The scheme is efficient, stable and scalable.

Overall, the methodology involves deriving an alternate representation for the presented loss function (and the associated class of nonsmooth loss functions) through a variational factorization process. This allows reformulation of problem objectives into a convenient, consistent form - which can then be optimized efficiently in a proximal block coordinate descent scheme. At the core, this involves solving non-linear L_1 minimization subproblems (nonlinear least absolute deviations), along with much simpler scalar convex ones. To avoid local minima, the whole optimization is carried with continuity under graduated nonconvexity.

For solving nonlinear least absolute deviations, a solver based on successive proximal minimization is proposed. It uses Peaceman - Rachford operator splitting for minimization, and can operate in a trust region framework if required. The approach is robust, efficient in practice, and has global convergence assurances. Least absolute deviations based approaches outperform their least squares and related counterparts in a variety of scenarios ([138, 139] for instance). Unfortunately, they are not easy to optimize - are particularly problematic in the nonlinear setting. Thus the solver is useful by itself.

5.1 Some notes on ρ - losses and influence functions of related estimators

Consider a loss function, $\rho : \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$, such as one shown in *Figure* 5.1a, or ones in *Figure* 5.2. For all practical purposes, when being utilized as in (5.8, 5.9, 5.10), the losses could be understood as weighting curves for the residues (specifically their norm, $\|\boldsymbol{r}\|$ or $|\boldsymbol{r}|$)

[1] Nonconvex objectives can have an exponential number of saddle points along with local optima, which makes even a proof of local optimality difficult (NP-hard) to construct.

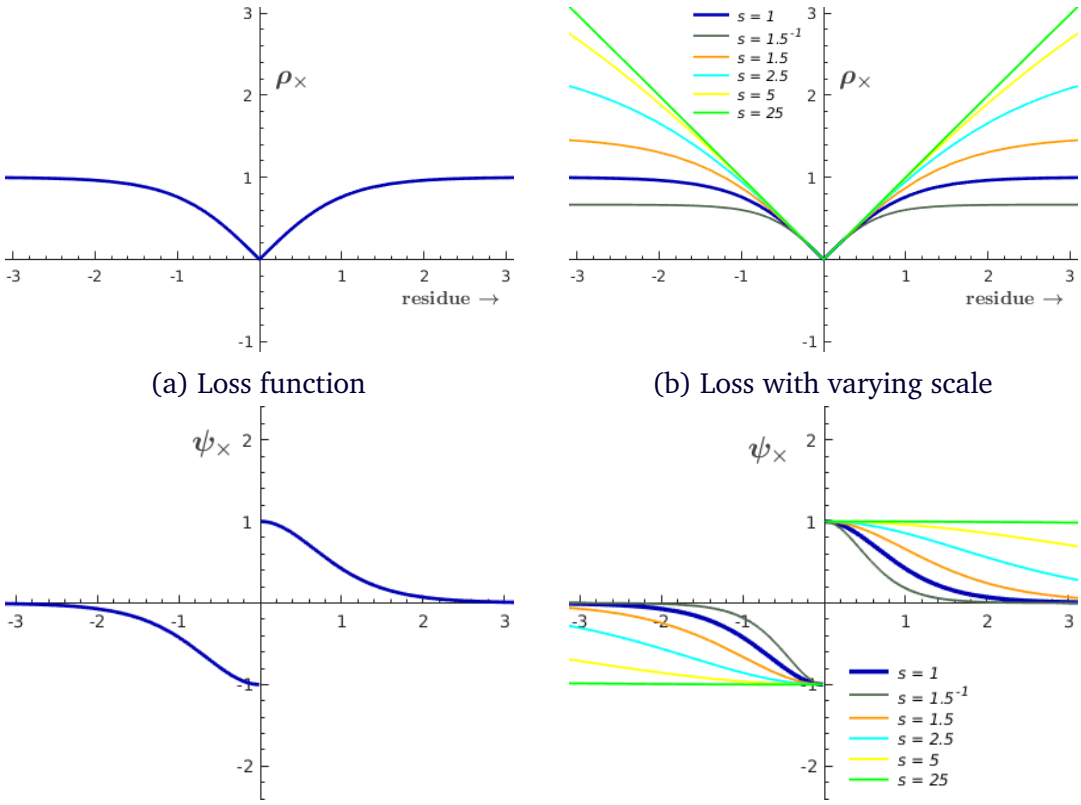
— basically regulating the response associated with each residue, and thereby leading to a particular θ estimate. For example, it can be seen that the L_2 loss (*Figure* 5.2a) attributes weights which grow quadratically, unchecked, with increasing residue values.

Besides analyzing the loss curve itself, useful insights on behavior of ρ can be obtained by observing its partial derivative with respect to residue, $\frac{\partial \rho}{\partial \boldsymbol{r}}$, denoted as ψ. The term captures the local sensitivity of ρ to perturbation in residue, and would arise when loss objectives (5.8, 5.9, 5.10 for instance), are being optimized — when their derivates with respect to θ are being evaluated for solving the first order optimality condition / constraint $\left(\frac{\partial E_{\cdot}}{\partial \theta_{\cdot}} = 0\right)$. It can then be seen that ψ is related to the bias / influence that a particular residue has on the solution, the θ - estimate.

The above characterization of ψ can be properly drawn out for estimation objectives of type indicated in (5.8, 5.9, 5.10), when the residual function is equivariant with θ — that is, when $\frac{\partial \boldsymbol{r}}{\partial \theta} \propto 1$. Then, ψ becomes directly proportional to the 'influence function' of ρ based estimators. The theory of influence functions, introduced by Hampel ([140]), characterizes how, in large samples / population, an infinitesimal proportion of data contamination affects the estimator (therefore its estimate).

Intuitively, the influence function is the change in an estimate caused by insertion of outlying data as a function of the distance of the data from the (uncorrupted) estimate. It captures the sensitivity of the estimate to a change / perturbation in observation of the estimate [2], and is hence is indicative of the estimator's local robustness. It is a measure of the amount of influence that a single perturbation can have over the estimate.

A related concept is that of the rejection point — it is defined as the point at which influence curve goes to zero. A finite rejection point protects against very large outliers. However, a finite rejection point ignores samples at tails of a distribution (for instance, *Figures* 5.2n and 5.2p). It thus usually results in the underestimation of scale (*Section* 5.2), adversely affects the statistical efficiency of the estimator [3], and may possibly give rise to additional local optima in an objective. For robust estimators especially, it is thus a good idea in general to utilize as many good samples (pertaining to the underlying distribution) as possible, in order to maintain good statistical efficiency.



(a) Loss function

(b) Loss with varying scale

(c) Influence function

(d) Influence with varying scale

Figure 5.1: $\rho_\times$: Proposed loss function $\rho_\times$

5.2 $\rho_\times$ and its properties

Equation 5.2 below gives the scaled version of the estimator, $\rho_\times(\boldsymbol{r};\mathcal{s})$ [4]. Its differential, $\psi_\times$, is given in (5.3).

$$\rho_\times(\boldsymbol{r};\mathcal{s}) = \mathcal{s}\frac{e^{2|\boldsymbol{r}|/\mathcal{s}}-1}{e^{2|\boldsymbol{r}|/\mathcal{s}}+1} \tag{5.2}$$

$$\psi_\times = \frac{\partial\rho_\times}{\partial\boldsymbol{r}} = \frac{4e^{2|\boldsymbol{r}|/\mathcal{s}}}{(e^{2|\boldsymbol{r}|/\mathcal{s}}+1)^2}\,\text{sign}(\boldsymbol{r}),\ \ \boldsymbol{r}\in\mathbb{R}^{d(\boldsymbol{r})}_{/\mathbf{0}} \tag{5.3}$$

Above, the sign function is taken element wise, as $\text{sign}(\boldsymbol{r}) \equiv \{\text{sign}(\lfloor\boldsymbol{r}\rfloor_c)\}_{c=1}^{d}$. Operation $d_{(..)}$ is used to retrieve dimensionality, and $\lfloor..\rfloor_c$ is used to retrieve the c^{th} dimension component. Some properties of $\rho_\times$ are noted in (5.4).

$$\lim_{|\boldsymbol{r}|\to\infty}\rho_\times = \mathcal{s} \tag{5.4a}$$

$$\sup_{\boldsymbol{r}}|\rho_\times| = \mathcal{s} \tag{5.4b}$$

$$\sup_{\boldsymbol{r}}|\psi_\times| = 1 \tag{5.4c}$$

$$\lim_{|\boldsymbol{r}|\to\infty}|\psi_\times| = 0 \tag{5.4d}$$

$$\arg\sup_{\boldsymbol{r}}|\psi_\times| = \mathbf{0} \tag{5.4e}$$

$$\frac{\partial\lfloor\psi_\times\rfloor_c}{\partial\lfloor\boldsymbol{r}\rfloor_c} < 0\ ,\forall c,\ \boldsymbol{r}\in\mathbb{R}^{d(\boldsymbol{r})}_{/\mathbf{0}} \tag{5.4f}$$

[2] For instance, in location estimation, with $residue = \theta - y$, the influence function would capture the sensitivity of the location estimate, θ, to a change in measurement, y.

[3] Efficiency is a measure of quality of an estimator. An estimator with good efficiency will have the variance of its estimate is as close as possible to the variance of the best estimator for a given distribution.

[4] Generally, scaled version of an estimator can be formulated as $\rho(\boldsymbol{r};\mathcal{s}) = \mathcal{s}^q\rho\left({}^{a}\boldsymbol{r}/_{\mathcal{s}}\right)$, when the estimator has L_p norm terms of the form $\|\boldsymbol{r}\|_p^q$. a is a tuning constant for maximizing statistical efficiency - it depends on the data distribution. The scaling constant $\mathcal{s}$ depends on the distribution and scale of the data in question, and is left up to the system designer. A popular approach to estimate it is through the median absolute deviation about the median (MADAM) $\to \mathcal{s} \coloneqq median\left\{|\boldsymbol{r}_i - median\left\{\boldsymbol{r}_i\right\}_{\forall i}|\right\}_{\forall i}/\mathcal{b}$. $\mathcal{b}$ is a correction constant for asymptotic consistency (0.6745 for normally distributed data when estimating the standard deviation)

$$\lim_{|\boldsymbol{r}|\to 0} \frac{\partial \lfloor \psi_\times \rfloor_c}{\partial \lfloor \boldsymbol{r} \rfloor_c} = 0 \ , \forall c \tag{5.4g}$$

Figure 5.1 plots the loss function and its influence curve. To compare, *Figure* 5.2 shows some varied loss functions which have figured more often in literature.

$\rho_\times$ is plotted in *Figure* 5.1a. It can be seen that $\rho_\times$ has a horizontal asymptote to ∞ (5.4a, 5.4b). This is needed in order to have a bounded response to outliers, and is only possible with nonconvexity or concavity. All (non-constant) convex losses, when their domains have not been truncated, have an unbounded sensitivity to gross errors / outliers, even in the presence of Tikhonov regularization ([141, 142]). Thus a single outlier is capable of skewing estimation arbitrarily when convex losses are used. This applies to all convex losses, including Huber ([143]) and L_1 (*Figure* 5.2c) [5] - estimates based on them can be arbitrarily off in face of large residual errors. Note that the Cauchy (also known as Lorentzian loss), although nonconvex, (*Figure* 5.2e) does not have a horizontal asymptote either.

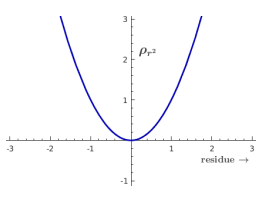
Figure 5.1c plots $\psi_\times$ (5.3), and is indicative of the influence function of $\rho_\times$. From *Figure* 5.1c and property (5.4c), it can be seen that the influence function is bounded. Thus $\rho_\times$ has bounded sensitivity - it is locally robust, with any (singular) perturbation only having a limited influence regardless of its magnitude.

A stronger, and global robustness property is that $\psi_\times$ descends to zero - property (5.4d). That is, any large observation errors have no influence on the estimate as they are completely suppressed - the loss function saturates. Such functions are called redescending estimators. Convex losses cannot have this property (*Figures* 5.2b and 5.2d). Clipped losses on the other hand, like the clipped L_2 and L_1 losses (*Figures* 5.2m through 5.2p), go abruptly and absolutely to zero (at a prespecified residue limit). This truncates tail influences altogether which, usually, significantly hampers their statistical efficiency (also look at discussion in *Section* 5.8).

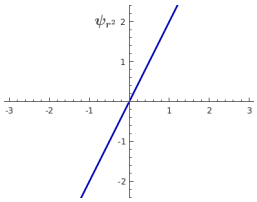
In comparison to the smooth redescenders in *Figure* 5.2, $\rho_\times$ rejects large outliers hardest. From (5.3), it becomes clear that $\psi_\times$ starts to aggressively suppresses influence for larger residues, when the denominator (growing exponentially faster) begins to dominate.

$\rho_\times$ is strictly concave in $|\boldsymbol{r}|$, $|\boldsymbol{r}| \in [0, \infty)$. Hence $\psi_\times$ is strictly decreasing, with zero residues having the highest influence (properties 5.4e and 5.4f). These are important properties - they imply that the perfect inliers (zero residues) have the most impact on estimation, and the influence progressively decreases with increase in residual error. Apart from $\rho_\times$ and Geman-Reynolds (*Figure* 5.2k), all the other losses in *Figure* 5.2 do not satisfy (5.4e) and

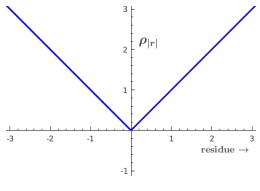
[5] For example, in learning linear regression parameters (θ, $residue = x^T\theta - y$), the L_1 loss is arbitrarily sensitive when there are errors in measurement (along y) as well as data (along x)



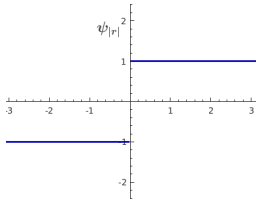
(a) L_2



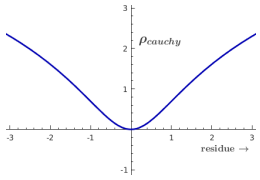
(b) L_2 ψ



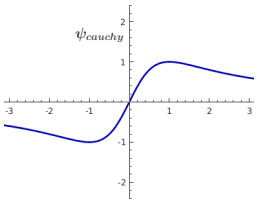
(c) L_1



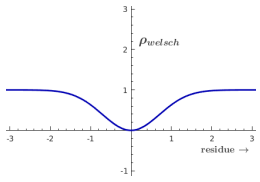
(d) L_1 ψ



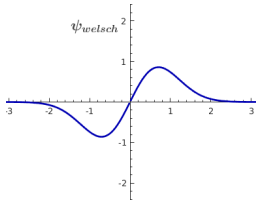
(e) Cauchy / Lorentzian



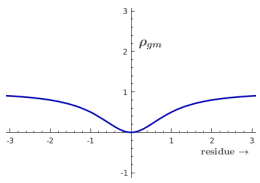
(f) Cauchy ψ



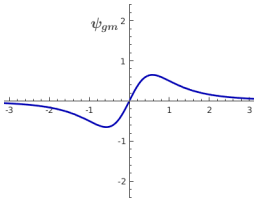
(g) Welsch / Leclerec



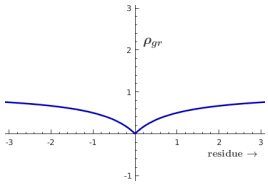
(h) Welsch ψ



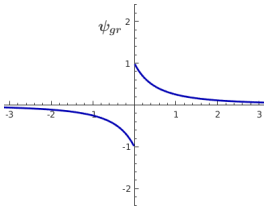
(i) Geman-Mclure



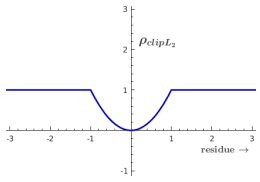
(j) Geman-Mclure ψ



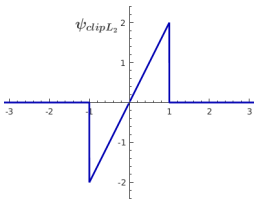
(k) Geman-Reynolds



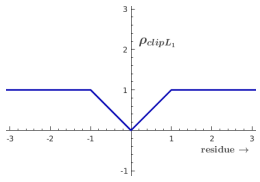
(l) Geman-Reynolds ψ



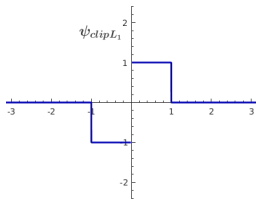
(m) Clipped L_2



(n) Clipped L_2 ψ



(o) Clipped L_1



(p) Clipped L_1 ψ

Figure 5.2: **Standard loss functions :** Some loss functions that have figured in perception literature

(5.4f) [6].

Property 5.4g, in conjuction with (5.4e) and (5.4f), is important as well. It says that near zero residues have similar influence in $\rho_\times$ estimates. $\psi_\times$ thus allows influence from zero or near zero residues to get through fully, before descending strongly and eventually suppressing high residual errors. Properties (5.4e) through (5.4g) together with (5.3) ensure that the best inliers (with zero or near zero residues) have maximal and similar impact on estimation, while the weaker inliers are significantly downweighted and thereon aggressively rejected. None of the losses in *Figure* 5.2 satisfy (5.4e) through (5.4g). Estimates from the smooth losses, such as *Figures* 5.2e through 5.2j), even though robust, would be biased significantly from weaker inliers and / or outliers; while estimate from a loss such as Geman - Reynolds (*Figures* 5.2k and 5.2l) would be unduly influenced by only a few samples with zero residues.

The nonsmoothness of $\rho_\times$ lends itself to another strong mathematical property ([144, 145, 146], the property applies to some other nonsmooth losses as well, such as ρ_{L_1}, but not to smoothed approximations of nonsmooth losses such as Huber, and clipped ones which are nonsmooth elsewhere such as clipped L_2). The nonsmoothness at zero with $\psi(0^-) < \psi(0^+)$, results in an estimate that exactly fits a potentially large subset of data. The estimate is also stable under weak data perturbations. In contrast, any smooth loss function ρ_{smooth}, with very high probability, will never exactly fit even a single datum (a more qualified statement is in the footnote [7]). This property is quite attractive for exact estimation (exactly fitting an inlier subset) in real world or realistic data (which is almost always noisy). Together with (5.4) this makes $\rho_\times$ well suited for estimation over data with low inlier ratios.

5.3 Variational Factorization

$$\rho(\boldsymbol{r}) \equiv \rho(\boldsymbol{r}|\mathring{\nu}) = \min_{0\leq\nu\leq 1} \rho(\boldsymbol{r}|\nu) \tag{5.5a}$$

$$\rho(\boldsymbol{r}|\nu) = q(\nu)\ell(\boldsymbol{r}) + p(\nu) \tag{5.5b}$$

To optimize $\rho_\times$ (and other related losses, objectives based on it), we first factorize it into

[6] In general, squared L_2 based noncovex losses (Geman-Mclure, Cauchy, Welsch for example, 5.2e through 5.2j) and clipped convex ones (5.2o for instance) will not satisfy 5.4e and 5.4f together

[7] Adapted from [144, 145, 146] — Assuming estimation of type referred to in Equation 5.8, with $r(\theta; o_i) = x_i^T\theta - y_i$ and $\rho(r)$ being sufficiently smooth in $\mathbb{R}_{/0}$. It can be shown that if $\psi(0^-) < \psi(0^+)$, then typical data $\{y_i, x_i\}_{\mathcal{D}}$ will give rise to local minimizers $\hat{\theta}$ of (5.8) which fit exactly a certain number of the data entries. There is a possibly large set, $\hat{h}_{exact}$, of data points such that $x_i^T\hat{\theta} = y_i\,, \forall i \in \hat{h}_{exact}$. Additionally, all strict local minimizers $\hat{\theta}$ are stable under weak pertubations of $\{y_i\}_{\mathcal{D}}$. In contrast, if $\rho(r)$ is smooth everywhere, for almost every set $\{y_i, x_i\}_{\mathcal{D}}$, the local minimizers of (5.8) do not fit any entry of $\mathcal{D}$. Thus, the possibility that a local minimizer fits some data entries is due to the nonsmoothness of $\rho(r)$

separately optimizable terms [8]. Equation 5.5 formulates the general form of the variational scheme we use. Here, $\ell(\boldsymbol{r})$ is a function which is closed, proper and convex in $\boldsymbol{r}$. It is typically a simple loss. $\nu \in [0, 1]$ is an auxillary variable introduced to enable the factorization. It couples $\ell(\boldsymbol{r})$, with an associated penalty, given by a closed, proper function $p(\nu)$ which is convex in $[0, 1]$. $\mathring{\nu}$ referes to the optimum value from the minimization in 5.5a. The coupling term, $q(\nu)$, is a function linear in ν over the support $[0, 1]$. The minimum of this family of functions parameterized by ν is the actual loss function, $\rho(\boldsymbol{r})$ (Equation 5.5, [142, 147, 149] for derivations). When $\ell(\boldsymbol{r})$ is the squared L_2 loss, clear connections can be drawn with generalized weiszfeld and other fixed point methods such as quasi-Newton, gradient linearization and reweighted least squares ([149, 150]). Note that this text focusses on nonsmooth losses (have $\ell(\boldsymbol{r})$ as $|\boldsymbol{r}|$)

The factorization process is quite powerful. A rather broad set of functions can be factorized through it ([142, 151, 147, 148]). Following analysis of Yu *et al.* ([142]), a variational representation can be derived for ρ, if and only if $\rho = a \cdot p^{f*} \circ (-\ell) + b$. Here $\circ$ indicates function composition, p^{f*} is the Fenchel conjugate of p (support of p is restricted to $\nu \in [0, 1]$), $a \,\&\, b$ are constants and ℓ is defined over all of the real number line.

We arrived upon $\rho_\times$ by following the reverse process, starting with particular ℓ and $\hat{p}$ having the desired properties and structure. Equation 5.6 gives the ν-relaxed factorization for $\rho_\times$.

$$\ell_\times(\boldsymbol{r}) = |\boldsymbol{r}|,\ q_\times(\nu) = \nu,\ p_\times(\nu) = \mathscr{s}\left(\sqrt{1-\nu} + \nu \log \frac{\sqrt{1-\nu}-1}{\sqrt{\nu}}\right) \tag{5.6a}$$

$$\rho_\times\left(\boldsymbol{r}\right) \equiv \rho_\times\left(\boldsymbol{r}|\mathring{\nu}\right) = \min_{0<\nu\leq 1} \rho_\times\left(\boldsymbol{r}|\nu\right) \tag{5.6b}$$

$$\rho_\times(\boldsymbol{r}|\nu) = \nu|\boldsymbol{r}| + \mathscr{s}\left(\sqrt{1-\nu} + \nu \log \frac{\sqrt{1-\nu}-1}{\sqrt{\nu}}\right) \tag{5.6c}$$

Equations 5.6b-5.6c can be interpreted as follows — for each (base) loss that is incurred in the absolute sense ($|\boldsymbol{r}|$, due to some residue $\boldsymbol{r}$), there is an associated importance weight (given by $\mathring{\nu}$) and penalty (given by $p_\times(\mathring{\nu})$), that regulate how much that residue contributes towards the net loss (5.8), and hence its impact / influence on the parameter estimation. Equation 5.15 shows how a residue and its importance weight are related for $\rho_\times$ - it has the same form as the influence curve, $\psi_\times$. $\mathring{\nu}$ thus serves as an explicit outlier / inlier marker — the closer it is to 1, the better the corresponding data point fits the estimated model, and

[8] The idea is to gainfully factorize the loss by introducing an auxilliary variable, ν. Firstly, to have resultant terms that can be effectively optimized separately, and secondly, to have an explicit characterization, through ν, of the sample importance weighting / outlier suppression property of the loss function. The later benefit extends beyond theoritical insights - one can add (possibly domain specifc) additional regularization / interaction terms based on ν, that can directly regulate the loss function characteristics in a desirable fashion ([147, 148, 142])

vice versa.

Note that any convex loss function ρ_{cvx} can be trivially represented in a ν-relaxed factorization. For instance, Equation 5.7 gives the trivially factorized form of L_1 loss. In general for any convex loss, ρ_{cvx}, we have $\ell_{cvx} = \rho_{cvx},\ q_{cvx}(\nu) = \nu$ and $p_{cvx}(\nu) = I_{[\nu=1]}$.

$$\ell_{L_1}(\boldsymbol{r}) = |\boldsymbol{r}|,\ q_{L_1}(\nu) = \nu,\ p_{L_1}(\nu) = I_{[\nu=1]} \tag{5.7a}$$

$$\rho_{L_1}(\boldsymbol{r}, \nu) = \nu|\boldsymbol{r}| + \infty I_{[\nu=1]} \tag{5.7b}$$

5.4 Robust Optimization

We optimize objectives of the forms modeled in Equations 5.8 through 5.10. Apart from $\rho_\times$, the analysis applies to a general class of nonsmooth, median type nonconvex loss functions. Such loss functions, as discussed in *Section* 5.2, are well suited for problem settings requiring robustness, sparsity or exact fitting / estimation.

The objective forms in Equations 5.8 through 5.10 figure prominently in modeling of overdetermined systems. They are optimized similarly (*Section* 5.5) - similar analysis should remain applicable on other objective forms as well.

$$E_1 = \sum_{i=1}^{N_{\mathcal{D}}} \rho\left(\boldsymbol{r}\left(\theta; o_i\right)\right) \tag{5.8}$$

$$E_2 = \sum_{g=1}^{N_\rho} \lambda_g \sum_{i=1}^{N_{\mathcal{D}}^g} \rho^g\left(\boldsymbol{r}^g\left(\theta_g; o_i^g\right)\right) \tag{5.9}$$

$$E_3 = \sum_{i=1}^{N_{\mathcal{D}}} \rho^0\left(\boldsymbol{r}^0\left(\theta_i; o_i\right)\right) + \sum_{g=1}^{N_\rho} \lambda_g \sum_{\{j,k\} \in C^g} \rho^g\left(\boldsymbol{r}^g\left(\theta_j, \theta_k; o_{j,k}^g\right)\right) \tag{5.10}$$

Equation 5.8 formulates the standard estimation and loss minimization objective. The form is known as M-estimation in statistics, and empirical risk minimization in machine learning literature. The generally applicable (5.8) appears in robust parameter estimation, regression, model fitting, subspace learning / estimation and several supervised learning problems to name a few. The model parameters are indicated by the vector, $\theta \in \mathbb{R}^{d(\theta)}$. As used earlier in the text, $\boldsymbol{r}\left(\theta; o_i\right) \in \mathbb{R}^{d(r)}$, is the residual error function (residue) which captures the error

in θ estimates. It could be the disparity between observations and predictions or a more involved function of the data/observations and the associated unknown parameterization. We use o_i to jointly refer to the data/observations. θ refers to the unknown parameterization to be estimated.

Equation 5.10 is a general formulation applicable to problems with structure (for instance, spatial, temporal or some other notion of proximity). The terms typically model regularization, apart from fidelity to data and observations. Equation 5.10 occurs, for instance, in tasks involving structurally regularized estimation, structured learning, multiple model fitting and regularized subspace learning. It occurs in various reconstruction and recovery tasks in vision and signal processing that require joint optimization and structural regularizations. The left term is a unary term - employed either to incorporate model priors, or to encourage fidelity to data or model measurements. The parameters of the model associated with the i^{th} datum are indicated as the vector, $\theta_i \in \mathbb{R}^{d(\theta_i)}$. The right term is a set of N_ρ summation terms for regularizing structural interactions pairwise, and to perhaps integrate any associated measurements / constraints, indicated singularly through $o^g_{j,k}$. Each kind of interaction is encoded by a respective set of edges, $C^g|_{g=1}^{N_\rho}$, over the models (their parameters) relevant to that interaction. The estimators / loss functions are distinguished through superscripts, $\rho^g|_{g=0}^{N_\rho}$. Correspondingly, $\boldsymbol{r}^g|_{g=0}^{N_\rho}$, with $\boldsymbol{r}^g \in \mathbb{R}^{d(\boldsymbol{r}^g)}$, are the associated residual functions. It is assumed that there is no overlap between the various parameter vectors, *i.e.* $\theta_i \cap \theta_j = \varnothing, \forall i \neq j$. λ_g are weighting constants balancing the various terms.

The form in Equation 5.9 subsumes the one in Equation 5.8, and occurs similarly in literature. Again there is no overlap between the various parameter vectors, *i.e.* $\theta_g \cap \theta_{g'} = \varnothing, \forall g \neq g'$. Apart from being more expressive than Equation 5.8, it admits a neat variable block structure - thus can explicitly leverage a block optimization scheme. Also note that Equation 5.10 can often be fully expressed in the form of Equation 5.9, and even Equation 5.8. Although not necessarily helpful in terms of leveraging blockwise optimization, the reformulation would admit a simpler sufficiency condition for convergence.

Instances in literature solving Equations 5.8 through 5.10 do so assuming a particular loss function, ρ. Convexity and / or smoothness of some of the terms is assumed. Generally, either *a)* L_2 based losses are employed, which result in least squares based subproblems and can leverage standard solvers off-the-shelf, or *b)* L_1 or similar loss is employed with residual functions linear / convex in parameters, so as to ensure overall convexity of the term. Various forms of strictly inferior relaxations and approximations of the original objective are commonly employed as well - like convex relaxations and smooth approximations of the problematic terms. Methodologies have also often employed very specific solution schemes - that are fundamentally tied with intricacies of the particular objective.

The ensuing analysis applies to a broad class of ν-factorizable nonsmooth, nonconvex loss

functions which have L_1 as the base loss, *i.e.* $\ell(\boldsymbol{r}) = |\boldsymbol{r}|$ for ρ in E_1, and $\ell_g(\boldsymbol{r}^g) = |\boldsymbol{r}^g|$ for $\rho_g,\ \forall g$ in E_2 and E_3. The various residual functions $\boldsymbol{r}..$, each defined over some parameter vector(s) $\theta..$, can be arbitrary but have to be differentiable. The setup thus allows flexibility and freedom to model problems more realistically.

To optimize E_1, E_2 and E_3, we first express them in their ν-relaxed forms. For E_1, we have simply taken $\rho = \rho_\times$. For E_2 and E_3, as mentioned earlier, we have assumed that the base losses for $\rho_g, \forall g$ are L_1. The coupling terms $q^g(\nu.), \forall g$ are taken simply as $\nu.$, without loss of generality. Equations 5.8 through 5.10 can then be rewritten as follows -

$$E_1 = \sum_{i=1}^{N_\mathcal{D}} \rho_\times \left(\boldsymbol{r}\left(\theta; o_i\right) | \mathring{\nu}_i\right) \tag{5.11a}$$

$$\rho_\times \left(\boldsymbol{r}\left(\theta; o_i\right) | \mathring{\nu}_i\right) = \min_{0<\nu_i\leq 1} \rho_\times \left(\boldsymbol{r}\left(\theta; o_i\right) | \nu_i\right) \tag{5.11b}$$

$$\rho_\times \left(\boldsymbol{r}\left(\theta; o_i\right) | \nu_i\right) = \nu_i |\boldsymbol{r}\left(\theta; o_i\right)| + \mathfrak{s}\left(\sqrt{1-\nu_i} + \nu \log \frac{\sqrt{1-\nu_i}-1}{\sqrt{\nu_i}}\right) \tag{5.11c}$$

$$E_2 = \sum_{g=1}^{N_\rho} \lambda_g \sum_{i=1}^{N_\mathcal{D}^g} \rho^g \left(\boldsymbol{r}^g\left(\theta_g; o_i^g\right) | \mathring{\nu}_i^g\right) \tag{5.12a}$$

$$\rho^g \left(\boldsymbol{r}^g\left(\theta_g; o_i^g\right) | \mathring{\nu}_i^g\right) = \min_{0\leq\nu_i\leq 1} \rho^g\big(\boldsymbol{r}^g\big(\theta_g; o_i^g\big) | \nu_i^g\big),\ \forall g \in \{1, \ldots N_\rho\} \tag{5.12b}$$

$$\rho^g\big(\boldsymbol{r}^g\big(\theta_g; o_i^g\big) | \nu_i^g\big) = \nu_i^g \big|\boldsymbol{r}^g\big(\theta_g; o_i^g\big)\big| + \boldsymbol{p}^g(\nu_i^g),\ \forall g \in \{1, \ldots N_\rho\} \tag{5.12c}$$

$$E_3 = \sum_{i=1}^{N_\mathcal{D}} \rho^0 \left(\boldsymbol{r}^0\left(\theta_i; o_i\right) | \mathring{\nu}_i^0\right) + \sum_{g=1}^{N_\rho} \lambda_g \sum_{\{j,k\}\in\mathcal{C}^g} \rho^g\big(\boldsymbol{r}^g\big(\theta_j, \theta_k; o_{j,k}^g\big) | \mathring{\nu}_{j,k}^g\big) \tag{5.13a}$$

$$\rho^0 \left(\boldsymbol{r}^0\left(\theta_i; o_i\right) | \mathring{\nu}_i^0\right) = \min_{0\leq\nu_i\leq 1} \rho^0 \left(\boldsymbol{r}^0\left(\theta_i; o_i\right) | \nu_i^0\right) \tag{5.13b}$$

$$\rho^0 \left(\boldsymbol{r}^0\left(\theta_i; o_i\right) | \nu_i^0\right) = \nu_i^0 \big|\boldsymbol{r}^0\left(\theta_i; o_i\right)\big| + \boldsymbol{p}^0(\nu_i^0) \tag{5.13c}$$

$$\rho^g\big(\boldsymbol{r}^g\big(\theta_j, \theta_k; o_{j,k}^g\big) | \mathring{\nu}_{j,k}^g\big) = \min_{0\leq\nu_{j,k}\leq 1} \rho^g\big(\boldsymbol{r}^g\big(\theta_j, \theta_k; o_{j,k}^g\big) | \nu_{j,k}^g\big),\ \forall g \in \{1, \ldots N_\rho\} \tag{5.13d}$$

$$\rho^g\big(\boldsymbol{r}^g\big(\theta_j,\theta_k;o^g_{j,k}\big)|\nu^g_{j,k}\big)=\nu^g_{j,k}\big|\boldsymbol{r}^g\big(\theta_j,\theta_k;o^g_{j,k}\big)\big|+p^g(\nu^g_{j,k}),\ \forall g\in\{1,\dots N_\rho\} \tag{5.13e}$$

Above $\mathring{\nu}_i$, $\mathring{\nu}^g_i$, $\mathring{\nu}^0_i$ and $\mathring{\nu}^g_{j,k}$ refer to the ν-minimizers from Equations 5.11b, 5.12b, 5.13b and 5.13d respectively. At times, when the usage is evident, we omit explicitly noting the dependence on parameters to reduce notation overload - so $\boldsymbol{r}(\theta;o_i)\equiv\boldsymbol{r}_i(\theta)\equiv\boldsymbol{r}_i$ and $\boldsymbol{r}^g(\theta_g;o^g_i)\equiv\boldsymbol{r}^g(\theta_g)\equiv\boldsymbol{r}^g_g$ and $\boldsymbol{r}^g(\theta_j,\theta_k;o^g_{j,k})\equiv\boldsymbol{r}^g(\theta_j,\theta_k)\equiv\boldsymbol{r}^g_{j,k}$.

5.5 Proximal block coordinate descent

E_1 is worked with first. E_2 and E_3 are optimized similarly, just with the possibility to split computation over several blocks (one pertaining to each parameter vector, θ..).

Below, just to take note of the nature of L_1 norm - it is dimensionally separable and additive. $\odot$ denotes the Hadamard (element-wise) product. $||$ is used to indicate the element-wise application of absolute value function on the matrix or vector.

$$|\boldsymbol{x}|\triangleq\sum_c|\lfloor\boldsymbol{x}\rfloor_c|,\ \boldsymbol{x}\in\mathbb{R}^{d(\boldsymbol{x})} \tag{5.14a}$$

$$|\boldsymbol{x}|+|\boldsymbol{y}|\equiv\left|\begin{array}{c}\boldsymbol{x}\\\boldsymbol{y}\end{array}\right|,\ \boldsymbol{x}\in\mathbb{R}^{d(\boldsymbol{x})},\ \boldsymbol{y}\in\mathbb{R}^{d(\boldsymbol{y})} \tag{5.14b}$$

$$M\coloneqq\begin{bmatrix}a&c\\b&d\end{bmatrix}\Longrightarrow M||=\begin{bmatrix}|a|&|c|\\|b|&|d|\end{bmatrix} \tag{5.14c}$$

$$M\coloneqq\begin{bmatrix}a&c\\b&d\end{bmatrix}\Longrightarrow \operatorname{sign}(M)=\begin{bmatrix}\operatorname{sign}a&\operatorname{sign}c\\\operatorname{sign}b&\operatorname{sign}d\end{bmatrix} \tag{5.14d}$$

$$M||\equiv\operatorname{sign}(M)\odot M \tag{5.14e}$$

Optimizing $\boldsymbol{E_1}$: The factorized forms in Equation 5.11 and Equation 5.13 enable an alternating minimization scheme. As can be seen, the base-loss ($\ell_\times$) and penalty terms ($\rho_\times(\nu_i)$) are easily separable. The parameter vector, θ, and outlier process variables, ν_i, can be thus be solved in separate blocks.

However the standard block coordinate descent may not converge here. For block coordinate descent to converge, all the subproblems either need to be solved to their respective unique global optimum, or the objective function needs to be atleast quasi convex and continuous

differentiable. Blockwise minimization can be unstable, unreliable with nonconvex objectives - especially when the problems are nonsmooth ([152, 153, 154]).

In our case, while $\mathring{\nu}_i$ can be evaluated uniquely (by construction, often in closed form), optimizing for θ makes block coordinate descent problematic - although simpler than the original ρ-objective, minimizing $|\boldsymbol{r}_i(\theta)|$ is still a nonsmooth, nonconvex problem.

Fortunately, a convergent scheme is possible by employing some additional machinery while solving the θ subproblem. Specifically, a globally majorizing first order surrogate function can be uniquely minimized for convergent descent, instead of the θ block itself. Together with an additional regularity condition [9] on the objective, this insures global convergence to a critical point ([152, 155]). In the scenario when global majorization cannot be ascertained, convergence properties and stability can still be held by minimizing a locally majorant / dominating, strongly convex first order surrogate of the block and ensuring sufficient descent ([156, 157, 152, 153, 158]).

Both the aforementioned scenarios are addressed while solving the θ subproblem. This is covered in *Section* 5.6.

Solving Equation 5.11b for $\mathring{\nu}_i$ - from $\frac{\partial \rho_{\times}(\boldsymbol{r}_i|\nu_i)}{\partial \nu_i}|_{\hat{\theta}} = 0$ we will have

$$\hat{\mathring{\nu}}_i = \frac{4e^{2|\hat{\boldsymbol{r}}_i|/\mathfrak{s}}}{(e^{2|\hat{\boldsymbol{r}}_i|/\mathfrak{s}}+1)^2},\ \forall i \tag{5.15}$$

where $\hat{\theta}$ is the incumbent estimate of the parameter vector, and $\hat{\boldsymbol{r}}_i \equiv \boldsymbol{r}(\hat{\theta}; o_i)$. The above can then be plugged in Equation 5.11a. We'll have then

$$\arg\min_{\theta} \sum_{i=1}^{N_{\mathcal{D}}} \rho\left(\boldsymbol{r}_i|\hat{\mathring{\nu}}_i\right) = \arg\min_{\theta} \sum_{i=1}^{N_{\mathcal{D}}} \hat{\mathring{\nu}}_i|\boldsymbol{r}\left(\theta; o_i\right)| + constant \tag{5.16a}$$

$$\hat{\theta}^{+} = \arg\min_{\theta} \sum_{i=1}^{N_{\mathcal{D}}} \hat{\mathring{\nu}}_i|\boldsymbol{r}\left(\theta; o_i\right)| \tag{5.16b}$$

which delineates our θ subproblem, after doing away with the constant terms. This can then be specified in a matrix form, with $W_R \in \mathbb{R}^{d(\boldsymbol{r})\cdot N_{\mathcal{D}} \times d(\boldsymbol{r})\cdot N_{\mathcal{D}}}$ and $R \in \mathbb{R}^{d(\boldsymbol{r})\cdot N_{\mathcal{D}}}$. e_i is the i^{th} standard basis in $\mathbb{R}^{N_{\mathcal{D}}}$. Operator $\otimes$ below is the Kronecker product. $\hat{\theta}^{+}$ indicates the updated θ estimate.

$$\hat{\theta}^{+} = \arg\min_{\theta} W_R\, R||(\theta) \tag{5.17a}$$

[9] The regularity condition is trivially satisfied by the E_1 and E_2 forms

$$W_R = \sum_{i=1}^{N_{\mathcal{D}}} \hat{\mathring{\nu}}_i e_i e_i^T \otimes \mathbf{1}_{d(\boldsymbol{r})} \; , \; R||(\theta) = \begin{pmatrix} |\boldsymbol{r}(\theta; o_1)| \\ \vdots \\ |\boldsymbol{r}(\theta; o_i)| \\ \vdots \\ |\boldsymbol{r}(\theta; o_{N_{\mathcal{D}}})| \end{pmatrix} \tag{5.17b}$$

Equation 5.16b / 5.17 is basically a weighted, nonlinear version of the least absolute deviations problem [10].

Section 5.6 covers solving (5.17) with a proximal algorithm - a general purpose solver is presented. The solver, $_{NL_1}$ / $_{NL_1-TR}$, operates iteratively, by successively minimizing tight approximations / surrogates of (5.17) at the solution iterates - with convergent descent. Importantly, the approximation model ($m_F(\xi; \vartheta)$, 5.30) and it's minimization (Algorithms $_{NL_1}$, $_{NL_1-TR}$) is devised in the manner mentioned a bit earlier in this section - to ensure convergent updates for block coordinate descent as well.

Thus note that, for purposes of optimizing E_1 *i.e.* for block wise optimization, (5.17) need not be minimized fully to a critical point - the θ block can be updated with the result of just a single iteration of $_{NL_1}$ (Equation 5.18a) or $_{NL_1-TR}$ (Equation 5.18b), which minimizes the (local or global) majorant at $\hat{\theta}$ ($m_F(\xi; \hat{\theta})$, 5.30) with sufficient descent.

$$\hat{\theta}^+ = NL_1(\ldots < ..itr_{_{NL_1}} = 1, .. >) \tag{5.18a}$$

$$\hat{\theta}^+ = NL_1 - TR(\ldots < ..itr_{_{NL_1-TR}} = 1, .. >) \tag{5.18b}$$

E_1 can thus be minimized optimized by cyclically updating the ν subproblems (5.15) and the θ subproblem (5.17a / 5.18) until convergence.

Optimizing E_2 : Objective 5.9 / 5.12 is optimized similarly. E_2 has a regular block structure, as there is no overlap between the various parameter vectors, $\theta_g \cap \theta_{g'} = \varnothing, \forall g \neq g'$, and the terms involving them are all mutually separable. Thus they can be optimized in separate blocks without impacting objective convergence ([155, 157, 158]) - in a similar fashion, with similar considerations, as the θ subproblem in E_1.

[10] We say least absolute deviations to indicate an overdetermined system

As earlier, for the ν^g.. blocks, from $\frac{\partial \rho^g\left(\boldsymbol{r}^g\left(\theta_g; o_i^g\right)|\check{\nu}_i^g\right)}{\partial \nu_i^g} = 0$, we will have

$$\left|\boldsymbol{r}^g\left(\hat{\theta}_g; o_i^g\right)\right| + \frac{\partial p^g(\nu_i^g)}{\partial \nu_i^g} = 0 \; \forall i, \; \forall g \tag{5.19}$$

ν_i^g would be the solution of Equation 5.19. $p^g(\nu_i^g)$ is the penalty term in the variational factorization of ρ^g (for generality, we had not assumed a specific ρ^g, only that $\ell_g(\boldsymbol{r}) = |\boldsymbol{r}^g|$, and $q_g(\nu) = \nu$). 5.19 is easy to solve by design - $p^g(\nu_i^g)$ is scalar and convex [11].

ν_i^g can then be plugged into 5.12a to optimize θ_g blocks. We will have then

$$\arg\min_{\theta_g} E_2 = \arg\min_{\theta_g} \sum_{g=1}^{N_\rho} \lambda_g \sum_{i=1}^{N_{\mathcal{D}}^g} \hat{\check{\nu}}_i^g \left|\boldsymbol{r}^g\left(\theta_g; o_i^g\right)\right| + constants \,, \; \forall g \in \{1, \ldots N_\rho\} \tag{5.20a}$$

$$\hat{\theta}_g^+ = \arg\min_{\theta_g} \sum_{g=1}^{N_\rho} \lambda_g \sum_{i=1}^{N_{\mathcal{D}}^g} \hat{\check{\nu}}_i^g \left|\boldsymbol{r}^g\left(\theta_g; o_i^g\right)\right| \,, \; \forall g \in \{1, \ldots N_\rho\} \tag{5.20b}$$

This can then be reorganized in as follows

$$\hat{\theta}_g^+ = \arg\min_{\theta_g} W_R^g \, R^g||(\theta_g) \,, \; \forall g \tag{5.21a}$$

$$W_R^g = \sum_{i=1}^{N_{\mathcal{D}}^g} \lambda_g \hat{\check{\nu}}_i^g e_i e_i^T \otimes \mathbf{1}_{d(\boldsymbol{r}^g)} \,, \; R^g||(\theta) = \begin{pmatrix} |\boldsymbol{r}^g(\theta_g; o_1)| \\ \vdots \\ |\boldsymbol{r}^g(\theta_g; o_i)| \\ \vdots \\ \left|\boldsymbol{r}^g(\theta_g; o_{N_{\mathcal{D}}^g})\right| \end{pmatrix} \,, \; \forall g \tag{5.21b}$$

Equation 5.21 is of the same form as (5.17), and would be updated similarly through a single iteration of NL_1 or $NL_1{-}TR$ (5.18).

E_2 can then minimized by updating the ν.. and θ.. blocks in a cyclic or essentially cyclic fashion until convergence. Also at each cycle of updates, randomly shuffling the blocks' update order helps to avoid poor local minima more effectively than a fixed cycling order.

A block optimization scheme is useful for scalability, efficiency and deployment in distributed setups.

[11] The ν subproblem often admits closed form solutions

Optimizing E_3 : The ν blocks are solved in much the same fashion as before. From $\frac{\partial \rho^{\cdot}(\boldsymbol{r}_{\cdot}^{\cdot}|\nu_{\cdot}^{\cdot})}{\partial \nu_{\cdot}^{\cdot}} = 0$, we will have

$$\left|\boldsymbol{r}^0\left(\hat{\theta}_i; o_i\right)\right| + \frac{\partial p^0(\nu_i^0)}{\partial \nu_i^0} = 0 \; \forall i \tag{5.22a}$$

$$\left|\boldsymbol{r}^g\left(\hat{\theta}_j, \hat{\theta}_k; o_{j,k}^g\right)\right| + \frac{\partial p^g(\nu_{j,k}^g)}{\partial \nu_{j,k}^g} = 0 \; \forall j,k, \; \forall g \in \{1, \dots N_\rho\} \tag{5.22b}$$

As earlier, $\mathring{\nu}_i^0$ and $\mathring{\nu}_{j,k}^g$ would be the respective solutions of (5.22a) and (5.22b).

Note that, although the various parameter vectors, θ_i, have no overlap amongst themselves, the terms involving them cannot be mutually decoupled.

Thus, one way to optimize 5.13 would be to solve all the parameter vectors jointly in a single Θ - block, $\Theta = \{\theta_i\}_{i=1}^{N_{\mathcal{D}}}$, and cycle it with the ν blocks. As before, from properties (5.14), all the residue absolute losses $|\boldsymbol{r}^0(\theta_i)|, \forall i \; \& \; |\boldsymbol{r}^g(\theta_j,\theta_k)|, \forall g \; \forall j,k$ would get stacked vertically, and the joint Θ - subproblem would have the same form as Equation 5.17, and would be optimized using (5.18). The scheme would thus have the same convergence assurances as E_1 (or E_2 for that matter).

Alternatively, each θ_i parameter vector could be optimized in a separate block (with concomitant benefits). But unlike optimizing (5.12), a θ_i block here would contain nonsmooth cross terms with some of the other parameter vectors ($|\boldsymbol{r}^g(\theta_i, \theta_j)|, \; \exists j \neq i$). For (majorization based) block descent to have certifiable convergence in this case (*i.e.* nonsmooth block with cross terms), a regularity condition has to be satisfied by the objective (5.10). Specifically, all blockwise minimizers of (5.10) should also be its stationary points ([159, 155]) [12]. Otherwise, convergence to only a block wise minimum can be assured.

In the blockwise case, the subproblem for θ_i block would be as follows :

$$\underset{\theta_i}{\arg\min}\, E_3 = \underset{\theta_i}{\arg\min}\, \hat{\mathring{\nu}}_i^0 \left|\boldsymbol{r}^0\left(\theta_i; o_i\right)\right| + \sum_{g=1}^{N_\rho} \lambda_g \sum_{\hbar_i} \hat{\mathring{\nu}}_{i,k}^g \left|\boldsymbol{r}^g\left(\theta_i, \hat{\theta}_k; o_{i,k}^g\right)\right| + constants \tag{5.23a}$$

$$\hat{\theta}_i^+ = \underset{\theta_i}{\arg\min}\, \hat{\mathring{\nu}}_i^0 \left|\boldsymbol{r}^0\left(\theta_i; o_i\right)\right| + \sum_{g=1}^{N_\rho} \lambda_g \sum_{\hbar_i} \hat{\mathring{\nu}}_{i,k}^g \left|\boldsymbol{r}^g\left(\theta_i, \hat{\theta}_k; o_{i,k}^g\right)\right| \tag{5.23b}$$

[12] The vice versa is always going to be true - all stationary points of a solution are always coordinate wise minimum as well. Also, the forms E_1 and E_2 satisfy the regularity condition by default, by virtue of having mutually decoupled nonsmooth blocks

This could then be reorganized as earlier :

$$\hat{\theta}_i^+ = \arg\min_{\theta_i} W_R^i \, R^i||(\theta_i) \, , \ \forall i \ \in \{1 \dots N_{\mathcal{D}}\} \tag{5.24a}$$

$$W_R^i = \nu_i^0 e_1 e_1^T \otimes \mathbf{1}_{d(\boldsymbol{r}^0)} + \sum_{t=1}^{|\hbar_i|} \lambda_{g_i^t} \nu_{i,\hbar_i^t}^{g_i^t} e_{1+t} e_{1+t}^T \otimes \mathbf{1}_{d(\boldsymbol{r}^{g_i^t})} \, , \ R^i||(\theta_i) = \begin{pmatrix} \boldsymbol{r}^0(\theta_i) \\ \boldsymbol{r}^{g_i^1}(\theta_i, \hat{\theta}_{\hbar_i^1}) \\ \vdots \\ \boldsymbol{r}^{g_i^t}(\theta_i, \hat{\theta}_{\hbar_i^t}) \\ \vdots \\ \boldsymbol{r}^{g_i^{|g_i|}}(\theta_i, \hat{\theta}_{\hbar_i^{|g_i|}}) \end{pmatrix} \tag{5.24b}$$

where $\hbar_i$, g_i are ordered arrays of indices. $\hbar_i = \left\langle k \,|\{i,k\} \in \mathcal{C}^g, \, \forall g \in ug_i \right\rangle$, $g_i = \left\langle g \,|\{i,k\} \in \mathcal{C}^g, \, \forall g \in ug_i \right\rangle$ and $|g_i| = |\hbar_i|$. Set of unique constraint functions regularizing θ_i is indexed by the set $ug_i = \left\{ g \,|\, \exists \ k \ s.t. \ \{i,k\} \in \mathcal{C}^g, \, \forall g \in \{1, \dots N_\rho\} \right\}$. R^i is a function of θ_i. It has incumbent estimates of all the other parameter vectors, $\hat{\theta}_t, \ \forall t \in u\hbar_i$ plugged in. Here $u\hbar_i = \left\{ k \,|\, \exists \ g \ s.t. \ \{i,k\} \in \mathcal{C}^g, \, \forall g \in \{1, \dots N_\rho\} \right\}$ denotes the set of all unique parameter vectors (identified through their index) which share a regularizer / 2-clique with θ_i.

Equation 5.24 is again in the familiar matrix form, and would be updated similarly through a single iteration of NL_1 or $NL_1\text{-}TR$ (5.18). The minimization of E_3 would be carried as earlier as well, by cycling through the block updates.

5.6 Nonlinear Least Absolute Deviations

$$\hat{\vartheta} = \arg\min_{\vartheta} E_F(\vartheta) = \arg\min_{\vartheta} W_F \ F||(\vartheta) \tag{5.25}$$

A successive proximal minimization scheme is proposed to find stationary points of nonlinear L_1 objectives of the type shown in Equation 5.25 in the overdetermined case [13]. Objectives of form (5.25) figure prominently in diverse domains, albeit under linearity or convexity

[13] The benefits of L_1 based optimization (like robustness, sparsity) over its L_2 counterparts have been well noted in literature. The minimization could pertain to a least absolute deviations problem, or a basis pursuit type one. Least absolute deviation problems arise in overdetermined systems ($m \geq n$), such as in sparse or robust estimation, regression and several learning problems as discussed earlier. Basis pursuit type problems arise in underdetermined systems such as ones that arise in compressive sensing applications, and problems pertaining to sparse signal recovery and codebook / dictionary learning ($m < n$). We focus on the overdetermined case here. Additional, often problem specific, regularization is needed to condition the problem in the underdetermined case

conditions on F ([160, 161] for instance). The framework of convexity can be too restrictive though. As has been well known by now, practical / real world problems can often benefit from more faithful, more accurate models afforded by nonlinear, nonconvex formulations.

Unfortunately, optimization becomes difficult as soon as one departs from both convexity and smoothness [14]. There have been a few approaches in more recent literature that have been effective in practice in the nonsmooth, nonconvex setting. Most of them assume separability of the objective into smooth and nonsmooth terms ([163, 164, 165, 166], and / or impose convexity or other similarly strong regularity on the nonsmooth part [160, 167, 161]. Approaches have assumed direct proximability of the nonsmooth terms [15] ([168, 165, 158]), or have optimized smooth approximations ([169]). There are also methods which have employed gradient sampling and quasi Newton methodology ([170, 171]).

Objective separability is not assumed here. It is assumed that the residue functions, $\boldsymbol{r}_{..}$ (nested in $|.|$, 5.17b,5.21b,5.24b), are all smooth, *i.e.* $F : \mathbb{R}^n \mapsto \mathbb{R}^m$ is continuously differentiable (while $F||$ is locally Lipschitz continuous). $\vartheta \in \mathbb{R}^n$ in (5.25). $W_F \in \mathbb{R}^{m \times m}$ is an arbitrary weight matrix with non-negative coefficients.

The method operates iteratively. At each step, a tight, strongly convex, but nonsmooth, local approximation of the objective function about the current solution estimate is derived. If majorization of the approximation model can be ascertained (requires a Lipschitz growth condition on F), it is minimized directly. Else the model is minimized adaptively in a trust region framework that ensures a sufficiently dominant approximation. Convergent descent is thereby ensured in either case. The approximation model's minimization is carried by an operator splitting approach (Peaceman - Rachford).

Note that the optimization can be viewed as an instance of majorization - minorization / successive upper bound minimization meta algorithm ([172, 173, 174]). It also has similarities with some trust region based methods ([175], although trust region approximation models / subproblems are always smooth). In each iteration, we are basically modeling a surrogate which bounds tightly from above, has the same first order behavior as the objective at the point of approximation, and minimizing it (with sufficient convergence towards a stationary point). It follows that the approach has global convergence assurances ([156, 176, 152]).

[14] Standard convex optimization methods are not directly applicable ([162]). Popular gradient based approaches, although effective in nonconvex problems, require objective smoothness. Their nonsmooth counterparts based on subgradients and cutting plane methods have been known to be non-robust and inefficient in practice. Bundle based methods, which are the most reliable of subdifferential based approaches, are primarily designed for the convex setting - even ascertaining local optimality requires convexity assumptions. They are not as effective in the nonconvex case. Approaches based on proximal operators, such as proximal point, iterative Bregman (method of multipliers), proximal gradient, alternating direction method of multipliers (split Bregman), are not directly applicable. Although robust and potentially efficient, they require convexity, operator proximability and make other assumptions depending on the operator. Sequential quadratic programming approaches assume objective smoothness

[15] A convex function $f(x)$ is proximable if the minimizer of $f(x) + \frac{1}{2\lambda}\|x - a\|_2^2$ can be obtained in closed form or easily

The scheme works well in practice - is efficient, can handle large problems and is stable.

Approximating E_F locally : We first develop the local approximation model for E_F. Since E_F is locally Lipschitz continious, a tight first order local approximation can always be devised - due to the continuity and existence of generalized directional derivatives. Generalized directional derivatives, $\nabla^{\circ}_{\hat{\mathbf{d}}}$, are the analogues of (standard) directional derivatives in the nonconvex, nonsmooth case. Generalized subdifferentials (for nonconvex functions) are defined through them. We define $\nabla^{\circ}_{\hat{\mathbf{d}}}$ below, as we use it later.

$$\nabla^{\circ}_{\hat{\mathbf{d}}} f(\mathbf{x}) = \lim_{\substack{y \to x \\ t \downarrow 0}} \inf \frac{f(\mathbf{x} + t\hat{\mathbf{d}}) - f(\mathbf{x})}{t}\,,\ \mathbf{x} + \hat{\mathbf{d}} \in C^0 \tag{5.26}$$

Keeping the above in mind, we first devise a first order local approximation of the objective E_F, in the vicinity of ϑ as follows

$$E_F(\vartheta \oplus \xi) \approx \tilde{E}_F(\xi; \vartheta) \tag{5.27a}$$

$$\tilde{E}_F(\xi; \vartheta) \coloneqq W_F\, F||(\vartheta) + W_F\, J_{\partial F||}(\vartheta)\, \xi \tag{5.27b}$$

$\tilde{E}_F$ is our first local approximation. It is parameterized by ξ [16] which delineates the vicinity. We have $\tilde{E}_F(0; \vartheta) = E_F(\vartheta)$. $\tilde{E}_F$ also approximates E_F in the first order behaviour about ϑ. By first order, it is simply implied that the approximation, $\tilde{E}_F$, has the same generalized directional derivatives (same directional behaviour) as E_F, at the point of approximation, ϑ.

$J_{\partial F||}$ refers to an element in $\partial F||(\vartheta)$, where $\partial F||(\vartheta)$ is the generalized jacobian at ϑ. It is defined as

$$\partial F||(\vartheta) \coloneqq conv\left\{A \in \mathbb{R}^{m \times n} | \, \exists\, \vartheta_t \subset \mathbb{R}^n_{/\Omega_{F||(\vartheta)}} s.t.\, \vartheta_t \to \vartheta \, and \, J_{F||}(\vartheta_t) \to A\right\} \tag{5.28}$$

where $\Omega_{F||(\vartheta)} \in \mathbb{R}^n$ is the set where $F||$ fails to be differentiable. The generalized jacobian (Clarke) of a locally Lipschitz continuous function, possibly nonconvex, is defined as the set constructed from the convex hull of all possible limits of jacobian matrices, $J_{F||}(\vartheta_t)$ at point(s) ϑ_t converging to ϑ. It is the generalization of the generalized subdifferential to vector valued functions.

We have that $diag(sign(F(\vartheta))) \cdot J_F(\vartheta) \in \partial F||(\vartheta)$. Plugging $J_{\partial F||}(\vartheta) \coloneqq diag(sign(F(\vartheta))) \cdot$

[16] Operator $\oplus$ is just used to indicate increment. For cartesian spaces, $\vartheta \oplus \xi \triangleq \vartheta + \xi$. For Lie groups, $\vartheta \oplus \xi \triangleq \vartheta e^{\hat{\xi}}$ is the retraction / exponential mapping. $\hat{\xi}$ here is the lie algebra isomorphic with the local coordinates $\xi \in \mathbb{R}^n$

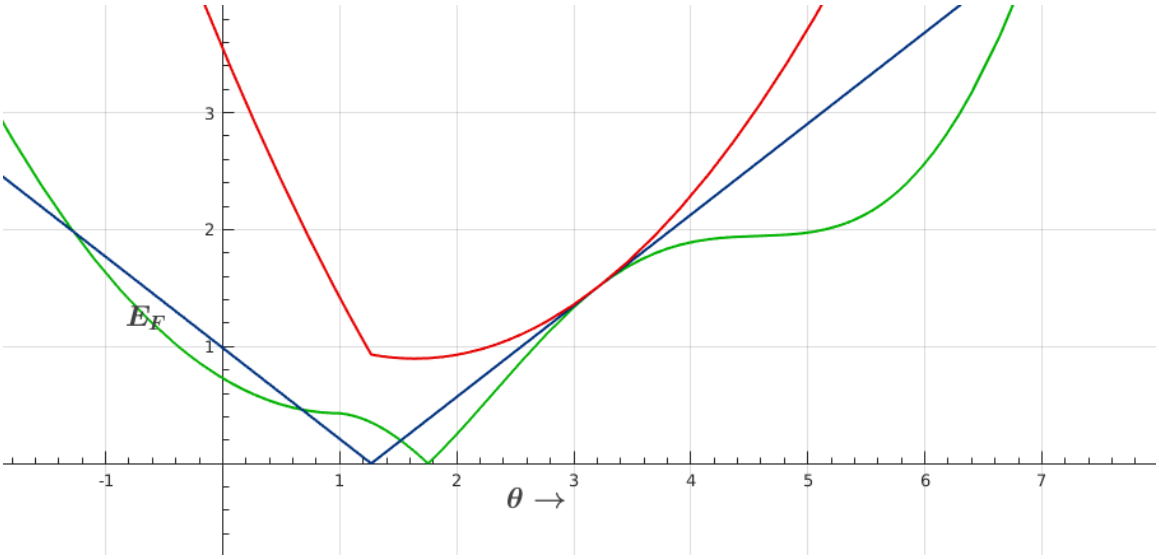


Figure 5.3: **Local approximation model :** The curve in green is E_F. The blue plot is the first order approximation, $\tilde{E}_F$ at $\vartheta \approx 3.22$. The red curve is the majorizing, strongly convex, local approximation model m_F. All curves in the figure are nonsmooth.

$J_F(\vartheta)$ in Equation 5.27, and approximating about $\hat{\vartheta}$, the incumbent ϑ estimate - we will have a concrete realization of the first order model (*Figure* 5.3) .

$$\tilde{E}_F(\xi;\hat{\vartheta}) = W_F|F(\hat{\vartheta}) + J_F(\hat{\vartheta})\xi| \tag{5.29}$$

We then add a quadratic term to $\tilde{E}_F(\xi;\hat{\vartheta})$ to get our final approximation model, $m_F(\xi;\hat{\vartheta})$.

$$m_F(\xi;\hat{\vartheta}) = W_F|F(\hat{\vartheta}) + J_F(\hat{\vartheta})\xi| + \frac{1}{2}\|\xi\|^2_{\Lambda_\zeta} \tag{5.30a}$$

$$\|\xi - \hat{\xi}^k\|^2_{\Lambda_\zeta} \triangleq (\xi - \hat{\xi}^k)^T\Lambda_\zeta(\xi - \hat{\xi}^k)\,,\ \Lambda_\zeta \succ 0,\ \Lambda_\zeta^T = \Lambda_\zeta \tag{5.30b}$$

where $\Lambda_\zeta = diag(\zeta_1, \ldots \zeta_n)$ makes the quadratic term, and hence $m_F(\xi;\hat{\vartheta})$, strongly convex (*Figure* 5.3). As mentioned earlier, $m_F(\xi;\hat{\vartheta})$ is minimized iteratively (in $TikhonovLAD{-}PRS$). The quadratic regularizer is important in multiple ways.

a) The quadratic term ensures strong convexity in m_F, which in turn ensures a unique and stable minimum. Uniqueness of the minimizer facilitates convergence of our block descent scheme.

$$c_{\Lambda_\zeta}\ :\ \xi \mapsto m_F(\xi;\vartheta) - \frac{1}{2}\|\xi\|^2_{\Lambda_\zeta} \tag{5.31a}$$

$$c_{\Lambda_\zeta} \text{ is convex } \forall \vartheta \implies m_F(\xi;\vartheta) \text{ is strongly convex with } \Lambda_\zeta \tag{5.31b}$$

b) The quadratic term averts divergence from a potential minima by bounding E_F from above. Under Lipschitz smoothness on F (quadratic growth at most) and accordingly lower bounded Λ_ζ, the quadratic term makes m_F a globally majorizing surrogate of E_F. m_F then upper bounds E_F, tightly from any approximation point ϑ in the effective domain, $dom\ E_F$. It also has the same directional gradients as E_F at the point of approximation. Specifically, the following are satisfied

$$m_F(\xi) \in \mathcal{C}^0 \tag{5.32a}$$

$$m_F(0;\vartheta) = E_F(\vartheta)\,,\ \forall \vartheta \in dom\, E_F \tag{5.32b}$$

$$\nabla^\circ_{\hat{\mathbf{d}}} m_F(0;\vartheta) = \nabla^\circ_{\hat{\mathbf{d}}} E_F(\vartheta)\,,\ \forall (\vartheta + \hat{\mathbf{d}}) \in dom\, E_F \tag{5.32c}$$

$$m_F(\xi;\vartheta) \geq E_F(\vartheta)\,,\ \forall (\xi \oplus \vartheta) \in dom\, E_F\,, \forall \vartheta \in dom\, E_F \tag{5.32d}$$

Given (5.32), minimization of $m_F(\xi;\vartheta)$ results in convergent descent. Successive minimization through Algorithm - $_{NL_1}$ is globally convergent, as it is an instance of majorization - minorization ([152, 156, 176]). The majorization also facilitates convergence for the block optimization procedure (Sections 5.5, [155])

c) In the scenario when Lipschitz growth condition on F and hence the last condition (5.32d) cannot be verified or satisfied, $m_F(\xi;\hat{\vartheta})$ is minimized in a trust region framework (Algorithm - $_{NL_1-TR}$). The trust region is adapted by regulating the quadratic term, through Λ_ζ. It ensures that minimizing $m_F(\xi;\hat{\vartheta})$ results in an acceptable descent step (υ^{accept} in $_{NL_1-TR}$, typically $\upsilon^{accept} \approx .2$). A stricter descent can be ensured as well, for global convergence assurances. m_F would then be adapted to sufficiently dominate locally — 5.33 below gives a sufficient condition for local majorization ([156, 166], together with 5.32a - 5.32c which m_F satisifies by design).

$$E_F(\hat{\vartheta} \oplus \hat{\xi}) \leq m_F(\hat{\xi};\hat{\vartheta})\,,\ \forall (\hat{\vartheta} \oplus \hat{\xi}) \in dom\, E_F \tag{5.33}$$

5.33 would be maintained by $_{NL_1-TR}$ when $\upsilon^{accept} \geq 1$. This also facilitates convergence of the block optimization procedure (Sections 5.5, [157, 158, 152])

d) Strong convexity and Lipschitz smoothness of the quadratic term also ensures a contractive mapping. The operator splitting approach (Peaceman - Rachford) used to minimize m_F requires atleast one of the operators to be contractive for objective convergence ([177, 178]). The minimizing fixed point iteration then converges strongly and geometrically.

Minimizing approximation model : Algorithm 5.1, NL_1, presents the procedure for minimizing 5.25. In each (outer, NL_1) iteration the approximation model, m_f is constructed and is passed on to the subroutine $TikhonovLAD{-}PRS$ for minimization.

To minimize m_F which is strongly convex but nonsmooth, we employ an approach based on proximal operations [17]. Specifically, an operator splitting scheme ([180]) [18] is applied over the dual of the approximation model, m_F, to find a fixed point minimizing m_F.

$$\min\ m_F(\xi;\hat{\vartheta}) \equiv \min_{\xi}\ W_F|F(\hat{\vartheta}) + J_F(\hat{\vartheta})\xi| + \frac{1}{2}\|\xi\|^2_{\Lambda_\zeta} \tag{5.34a}$$

$$\Leftrightarrow \quad \min\ h(\boldsymbol{x}) + g(\boldsymbol{z}) \ \text{ with } \boldsymbol{x} = B\boldsymbol{z} - \boldsymbol{c} \tag{5.34b}$$

$$\boldsymbol{z} = \xi,\ \boldsymbol{c} = -W_F F(\hat{\vartheta}),\ B = W_F J_F(\hat{\vartheta}),\ Q = \Lambda_\zeta \tag{5.34c}$$

$$h(\boldsymbol{x}) = |\boldsymbol{x}|,\ g(\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{z}\|^2_Q \tag{5.34d}$$

Above, the objective $\min\ m_F(\xi;\hat{\vartheta})$ (5.34a) is reformulated to (5.34b), by introducing an additional auxillary variable $\boldsymbol{x}$. The dual problem of (5.34b) is noted in Equation 5.35a below. h^* and g^* are the Fenchel conjugates as earlier. $\boldsymbol{y}$ is the dual variable, which figures as the Lagrange multiplier in the constrained primal. Strong duality holds due to convexity of m_F.

$$\min\ h^*(\boldsymbol{y}) + g^*(-B^T\boldsymbol{y}) - c^T\boldsymbol{y} \tag{5.35a}$$

[17] Proximal algorithms have become popular in recent years for convex optimization due to their general applicability and ability to handle high dimensional and large scale and distributed problems. They are well suited for nonsmooth analysis - are often free of derivatives altogether. "They sit at a higher level of abstraction than classical algorithms like Newton's method : the base operation is evaluating the proximal operator of a function" - [179]. The proximal operation, which lends itself to several interpretations, is a (significantly easier) convex optimization problem itself. The approach not just affords significant flexibility. It also provides a unifying framework for analyzing, and potentially extending, a wide ranging set of convex optimization methods

[18] An operator is a point to set mapping or a relation - it generalizes the notion of a function, and provides a unifying analytical construct usefully abstracted from encumbering problem specifics. Operator splitting methods work by separately optimizing the different terms in the objective in a fixed point iteration. A strongly (or weakly) convergent splitting scheme involves a sequence of operations which are contractive (or firmly nonexpansive) on the whole

Algorithm 5.1: *Nonlinear L_1 Solver* [$\boldsymbol{NL_1}$]

Function $Main_{NL_1}$

Input : $F(\vartheta)$, W_F , $< Input - Parameters >$

Output : $\hat{\vartheta}$

Input-Parameters : $\zeta_c \overset{d}{\leftarrow} \zeta_{high}\ \forall c \in \{1 \dots n\}$, $itr_{NL_1} \overset{d}{\leftarrow} \mathsf{t}_{high}$, $\mathsf{t}_{ProxL} \overset{d}{\leftarrow} \mathsf{t}_{high}$, $\vartheta_{init} \overset{d}{\leftarrow} 0$

// $\overset{d}{\leftarrow}$ indicates fallback / default value. $\cdot\cdot_{high}$ implies sufficiently high.

$\hat{\vartheta} \leftarrow \vartheta_{init}$

$\mathsf{t} \leftarrow 0$

While $\mathsf{t} < itr_{NL_1}$ & $!converged$

$\mathsf{B} \leftarrow W_F J_F(\hat{\vartheta})$

$\mathsf{c} \leftarrow -W_F F(\hat{\vartheta})$

$\mathsf{z}_{init} \leftarrow 0$

$\mathsf{Q} \leftarrow diag(\zeta_1, \dots \zeta_n)$

$\hat{\xi} \leftarrow$ `TikhonovLAD-PRS(` B , c , $< \mathsf{z}_{init}$, Q , $\mathsf{t}_{ProxL} >$`)`

$\hat{\vartheta} \leftarrow \hat{\vartheta} \oplus \hat{\xi}$

$\mathsf{t} \leftarrow \mathsf{t} + 1$

End

/* Contractive Peaceman-Rachford for $\arg\min_z |Bz - c| + \frac{1}{2}\|z\|_Q^2$ */

Function `TikhonovLAD-PRS`

Input : B , c , $< Input - Parameters >$

Output : z

Input-Parameters : $\mathsf{x}_{init} \overset{d}{\leftarrow} 0$, $\mathsf{Q} \overset{d}{\leftarrow} \zeta_{high}\mathbf{1}_n$, $\mathsf{t}_{ProxL} \overset{d}{\leftarrow} \mathsf{t}_{high}$, $\alpha \overset{d}{\leftarrow} .75$, $\beta \overset{d}{\leftarrow} .25$

// $\min |\mathsf{x}| + \frac{1}{2}\|\mathsf{z}\|_{\mathsf{Q}}^2$ with $\mathsf{x} = \mathsf{Bz} - \mathsf{c}$

$k \leftarrow 0$

$\mathsf{z}^k \leftarrow 0$

$\mathsf{x}^k \leftarrow \mathsf{x}_{init}$

$\mathsf{y}^k \leftarrow 0$

While $k < \mathsf{t}_{ProxL}$ & $!converged$

$\mathsf{z}^{k+1} = (\mathsf{Q} + \varrho \mathsf{B}^T \mathsf{B})^{-1}(\varrho \mathsf{B}^T(\mathsf{c} + \mathsf{x}^k - \mathsf{y}^k/\varrho)$ // $g(\mathsf{z}) \triangleq \frac{1}{2}\|\mathsf{z}\|_{\mathsf{Q}}^2$

$\mathsf{y}^{k+1/2} = \mathsf{y}^k + \alpha\varrho(\mathsf{Bz}^{k+1} - \mathsf{x}^k - \mathsf{c})$

$\mathsf{x}^{k+1} = shrink_{\frac{1}{\varrho}}(\mathsf{Bz}^{k+1} + \mathsf{y}^{k+1/2}/\varrho - \mathsf{c})$ // $h(\mathsf{x}) \triangleq |\mathsf{x}|$

$\mathsf{y}^{k+1} = \mathsf{y}^{k+1/2} + \beta\varrho(\mathsf{Bz}^{k+1} - \mathsf{x}^{k+1} - \mathsf{c})$

$k \leftarrow k + 1$

End

Algorithm 5.2: *Nonlinear L_1 Solver with Trust Region* $[\ \boldsymbol{NL_1 - TR}\ ]$

Function $Main_{NL_1-TR}$

Input : $F(\vartheta)$, W_F , $< Input - Parameters >$

Output : $\hat{\vartheta}$

Input-Parameters : $\zeta \overset{d}{\leftarrow} \zeta_{high}$, $\delta_0 \overset{d}{\leftarrow} 1$, $\Upsilon^{accept} \overset{d}{\leftarrow} 1$, $\Upsilon^{good} \overset{d}{\leftarrow} 1.5\Upsilon^{accept}$, $\delta^{inc} \overset{d}{\leftarrow} 2$, $\delta^{dec} \overset{d}{\leftarrow} .9$, $itr_{NL_1-TR} \overset{d}{\leftarrow} \mathsf{t}_{high}$, $\mathsf{t}_{ProxL} \overset{d}{\leftarrow} \mathsf{t}_{high}$, $\vartheta_{init} \overset{d}{\leftarrow} 0$

// $\overset{d}{\leftarrow}$ indicates fallback / default value. $\cdots_{high}$ implies sufficiently high.

$\hat{\vartheta} \leftarrow \vartheta_{init}$

$\mathsf{t} \leftarrow 0$

While $\mathsf{t} < itr_{NL_1-TR}$ & $!converged$

$\mathsf{B} \leftarrow W_F J_F(\hat{\vartheta})$

$\mathsf{c} \leftarrow -W_F F(\hat{\vartheta})$

$\mathsf{z}_{init} \leftarrow 0$

$\Lambda = \mathbf{1}_n \odot \mathsf{B}^T\mathsf{B}$

$\mathsf{Q} \leftarrow \zeta \frac{\Lambda}{\|\Lambda\|_2} \delta_{\mathsf{t}}^{-1}$ // $\|\Lambda\|_2 = \max_c\{\Lambda_{c,c}\}$

$\hat{\xi} \leftarrow$ `TikhonovLAD-PRS(` B `,` c `,` $< \mathsf{z}_{init}$ `,` Q `,` $\mathsf{t}_{ProxL} >$`)`

$\Upsilon_{\mathsf{t}} \leftarrow (E_F(\hat{\vartheta}) - E_F(\hat{\vartheta} \oplus \hat{\xi})) / (E_F(\hat{\vartheta}) - m_F(\hat{\xi}; \hat{\vartheta}))$

switch Υ_{t}

case $\Upsilon_{\mathsf{t}} \geq \Upsilon^{good}$

$\delta_{\mathsf{t}+1} \leftarrow \delta^{inc} \cdot \delta_{\mathsf{t}}$

$\hat{\vartheta} \leftarrow \hat{\vartheta} \oplus \hat{\xi}$

case $\Upsilon_{\mathsf{t}} < \Upsilon^{accept}$

$\delta_{\mathsf{t}+1} \leftarrow \delta^{dec} \cdot \delta_{\mathsf{t}}$

otherwise

$\delta_{\mathsf{t}+1} \leftarrow \delta_{\mathsf{t}}$

$\hat{\vartheta} \leftarrow \hat{\vartheta} \oplus \hat{\xi}$

$\mathsf{t} \leftarrow \mathsf{t} + 1$

End

$$\mathbf{0} \in \mathcal{F}(\boldsymbol{y})\,,\;\; \mathcal{F} = \partial h^*(\boldsymbol{y}) - B\partial g^*(-B^T\boldsymbol{y}) - c \tag{5.35b}$$

$$\mathbf{0} \in \mathcal{F}(\boldsymbol{y}) \;\Leftrightarrow\; \mathbf{0} \in (\mathcal{F}_h + \mathcal{F}_g)(\boldsymbol{y}) \tag{5.35c}$$

$$\mathcal{F}_h = \partial h^*(\boldsymbol{y}) - c,\, \mathcal{F}_g = -B\partial g^*(-B^T\boldsymbol{y}) \tag{5.35d}$$

Equation 5.35b states the dual problem as an equivalent generalized equation with operator $\mathcal{F}$. Here $\mathcal{F}$ is the subdifferential operator (standard subdifferential as m_F is convex) - so (5.35b) basically says that the subdifferental set at a stationary point $\boldsymbol{y}$ should necessarily contain a zero subgradient.

5.35c and 5.35d then indicate how $\mathcal{F}$ is split into two separate operations $\mathcal{F}_h$ and $\mathcal{F}_g$, giving way to the split generalized equation (5.35c). The split equation can then be effectively solved by the fixed point equation system 5.36a below.

$$C_{\mathcal{F}_h} \circ C_{\mathcal{F}_g}(\boldsymbol{w}) = \boldsymbol{w}\,,\;\; \boldsymbol{y} = R_{\mathcal{F}_g}(\boldsymbol{w}) \tag{5.36a}$$

$$R_{\mathcal{F}_f} \triangleq (I + \varrho\mathcal{F}_f)^{-1}\,,\, C_{\mathcal{F}_f} \triangleq 2R_{\mathcal{F}_f} - I \tag{5.36b}$$

$$R_{\mathcal{F}_f}(\boldsymbol{w}) \overset{\mathcal{F}_f=\partial f}{=\!=} prox_{\varrho f}(\boldsymbol{w}) \triangleq \arg\min_{\boldsymbol{t}} f(\boldsymbol{t}) + \frac{1}{2\varrho}\|\boldsymbol{t} - \boldsymbol{w}\|^2\,,\;\; \varrho > 0 \tag{5.36c}$$

The fixed point system 5.36a involving the composited operators and operations is orginally due to Peaceman and Rachford ([181]). The composite operators, cayley ($C_{\mathcal{F}_f}$), and resolvent ($R_{\mathcal{F}_f}$) are defined in (5.36b,5.36c) for an arbitrary convex function f. The Peaceman - Rachford splitting (PRS) above converges provably when atleast one of the cayley operators ($C_{\mathcal{F}_h}$, $C_{\mathcal{F}_g}$) is contractive. As g is strongly convex (5.34d), this is the case here with $C_{\mathcal{F}_g}$ — the system (5.36a) then converges strongly and faster than any of its damped variants (*Table* 5.1, [177, 182, 178]) [19].

Unrolling (5.36) and applying it to (5.34b), gets us the fixed point system (5.37) which

[19] The prominent Douglas-Rachford splitting is a damped version of PRS - the damping allows weak convergence under more relaxed conditions. Alternating direction method of multipliers (ADMM) is an example of Douglas-Rachford splitting.

minimizes m_F in the primal-dual.

$$\boldsymbol{z}^{k+1} = (Q + \varrho B^T B)^{-1}(\varrho B^T(\boldsymbol{c} + \boldsymbol{x}^k - \boldsymbol{y}^k/\varrho) \tag{5.37a}$$

$$\boldsymbol{y}^{k+1/2} = \boldsymbol{y}^k + \varrho(B\boldsymbol{z}^{k+1} - \boldsymbol{x}^k - \boldsymbol{c}) \tag{5.37b}$$

$$\boldsymbol{x}^{k+1} = shrink_{\frac{1}{\varrho}}(B\boldsymbol{z}^{k+1} + \boldsymbol{y}^{k+1/2}/\varrho - \boldsymbol{c}) \tag{5.37c}$$

$$\boldsymbol{y}^{k+1} = \boldsymbol{y}^{k+1/2} + \varrho(B\boldsymbol{z}^{k+1} - \boldsymbol{x}^{k+1} - \boldsymbol{c}) \tag{5.37d}$$

Above, $shrink$ is element-wise soft thresholding / shrinkage operation. It is the proximal operation over L_1 norm — $prox_{1/\varrho|\cdot|}(\boldsymbol{w}) \equiv shrink_{1/\varrho}(\boldsymbol{w})$. It is defined as

$$shrink_{1/\varrho}(\boldsymbol{w}) = sign(\boldsymbol{w}) \odot \{max(|\lfloor \boldsymbol{w} \rfloor_t| - 1/\varrho, 0)\}_{t=1}^{d(\boldsymbol{w})} \tag{5.38}$$

Two constants, α and β, are then added to 5.37b and 5.37d respectively, to regulate step sizes, for faster and robuster convergence ([183, 184, 185]). In practice, this translates into signifcantly faster convergence for particular (application specific) choices for α and β ([184] presents a recent analysis for allowable range of values).

$$\boldsymbol{z}^{k+1} = (Q + \varrho B^T B)^{-1}(\varrho B^T(\boldsymbol{c} + \boldsymbol{x}^k - \boldsymbol{y}^k/\varrho) \tag{5.39a}$$

$$\boldsymbol{y}^{k+1/2} = \boldsymbol{y}^k + \alpha\varrho(B\boldsymbol{z}^{k+1} - \boldsymbol{x}^k - \boldsymbol{c}) \tag{5.39b}$$

$$\boldsymbol{x}^{k+1} = shrink_{\frac{1}{\varrho}}(B\boldsymbol{z}^{k+1} + \boldsymbol{y}^{k+1/2}/\varrho - \boldsymbol{c}) \tag{5.39c}$$

$$\boldsymbol{y}^{k+1} = \boldsymbol{y}^{k+1/2} + \beta\varrho(B\boldsymbol{z}^{k+1} - \boldsymbol{x}^{k+1} - \boldsymbol{c}) \tag{5.39d}$$

The system 5.39 above is iterated till a minimizing fixed point is attained. Since m_F is strongly convex, the fixed point is its unique minimizer. The procedure is outlined as $TikhonovLAD{-}PRS$ in NL_1. Note that $TikhonovLAD{-}PRS$ involves a sequence of simple, exact operations which are efficient, can scale very well and are very stable.

Table 5.1: **TikhonovLAD−PRS evaluation over linear recovery task :** Average iterations required to sufficiently recover a source signal, from its corrupted linear encoding, are indicated. This was done for source signals of increasing length (wordsize). Different Lower values are better. $Q = \zeta\ \mathbf{1}_n$ for the experiment (*Function* $TikhonovLAD-PRS$). The results indicate graceful scaling. Note that when $\alpha = 0$, the method essentially corresponds to alternating direction method of multipliers ($ADMM$). Significant improvements in convergence can be achieved over it, as the results show.

Constants ↓ Wordsize →	***256***	***512***	***1024***
$\boldsymbol{\zeta = 0}$			
$\langle \alpha = 0, \beta = 1 \rangle$ $(ADMM)$	99.96	132.94	179.02
$\langle \alpha = .3, \beta = .7 \rangle$	94.27	123.64	171.76
$\langle \alpha = .7, \beta = .3 \rangle$	89.92	119.44	165.51
$\boldsymbol{\zeta = 5}$			
$\langle \alpha = 0, \beta = 1 \rangle$ $(ADMM)$	102.7	154.43	180.86
$\langle \alpha = .3, \beta = .7 \rangle$	94.72	123.18	157.44
$\langle \alpha = .7, \beta = .3 \rangle$	90.58	118.93	153.41
$\boldsymbol{\zeta = 10}$			
$\langle \alpha = 0, \beta = 1 \rangle$ $(ADMM)$	116.64	144.31	188.1
$\langle \alpha = .3, \beta = .7 \rangle$	104.20	129.34	175.74
$\langle \alpha = .7, \beta = .3 \rangle$	100.19	125.34	167.51

Minimizing approximation model with trust region : Trust region variant of the approach is presented in Algorithm 5.2, $NL_1 - TR$. A trust region framework is necessary when m_F majorization, and hence sufficient descent and convergence is not assured anymore.

$NL_1 - TR$ proceeds quite similarly, except that the quadratic regularizer, regulated by $Q = \Lambda_\zeta$, now gets adapted each (outer) iteration. This is done depending on the fitness of the current descent step ($\hat{\xi}$). The fitness score, indicated by Υ_t, basically captures the ratio of actual descent to the expected one, as predicted by the approximation model, m_F. A low fitness ($\Upsilon_t < \Upsilon^{accept}$) implies that there was insufficient descent. This would be because the approximation model failed to remain faithful, and hence a smaller region needs to be trusted. No update is made in this case, and the trust region (delineated by $Q \equiv \Lambda_\zeta$) is contracted by increasing the regularization (through δ_t). Similarly, more region is trusted when there is good descent. For a better descent step, Λ_ζ is diagonally reshaped in accordance with the (rough, Gauss-Newton) local curvature shape estimate ($\propto (\mathbf{1}_n \odot J_F(\hat{\vartheta})^T W_F^T W_F J_F(\hat{\vartheta}))/\|\mathbf{1}_n \odot J_F(\hat{\vartheta})^T W_F^T W_F J_F(\hat{\vartheta})\|_2$).

Although sufficient descent is insured by the aforediscussed scheme, a stricter local majorization condition is necessary for convergence in the nonconvex, nonsmooth case - (5.33) needs to be satisfied in order to ensure that the approximation model m_F dominates E_F locally. This can achieved by setting $\Upsilon^{accept} \geq 1$. Condition (5.33, [156]) basically makes sure that that the descent step is always bounded from below by the objective - E_F is lower than (not

above) m_F at the updated solution point - thus ensuring that the step / update does not diverge away from a potential minima in E_F. Similarly motivated conditions have have been employed in different settings [186, 160, 158].

Figure 5.4: **Performance curves :** We show the impact of optimizer iterations / lowest scale optimized, on accuracy of SE(3) estimates using the proposed loss, under graduated nonconvexity (*Section* 5.7). The noncovexity was regulated by varying the scale parameter, $\mathfrak{s}$, in 5.2 (more nonconvex at lower scales). $\mathfrak{s}$ was reduced by a constant factor every five iteration cycles (5.15, 5.17 / 5.18). The horizontal axis labels indicate the number of iterations and lowest scale optimized. The scale values are in metres and normalized point clouds were used in the experiment. Average translation errors, rotation errors and estimation success rates have been indicated in the top, middle and bottom plots respectively. The curves are from different data sets, with their average result being indicated by the thicker yellow curve. The results on right were evaluated on significantly noisier data (more outliers, marked as '$step25$') than the result plots on left (marked as '$step10$').

5.7 Tackling Local Minima

As is the case with nonconvexity, finding a global minimizer is NP-Hard. The focus then lies on finding good locally optimal solutions.

Graduated NonConvexity — The scale parameter $\mathfrak{d}$ regulates the convexity of $\rho_\times$ about the zero set (minimizing solutions, with zero residues) in a continuous manner. *Figure* 5.1b shows $\rho_\times$ with varying values of $\mathfrak{d}$ (similar analysis applies to other robust losses as well). It can be seen that with increasing scales, $\rho_\times$ behaves like strictly convex L_1 loss in the (correspondingly larger) neighborhoods about zero. Minimizing at a sufficiently high scale (akin to to solving an L_1 convex relaxation) thus results in a globally optimal solution, albeit with all data as inlier and affecting the solution estimate accordingly. The inferior solution estimate can serve to nicely initialize a more nonconvex / less relaxed $\rho_\times$, and would result in its refinement. This could be pursued successively while maintaining continuity of the objective through, $\mathfrak{d}$, and keeping track of the solution ([187]).

In practice, the technique is quite effective in avoiding bad local minima - can even help reach globally optimal solutions of the original nonconvex objective in certain cases ([188, 189]). It performed well in our experiments (*Figure* 5.4).

5.8 Experiments

Figure 5.5 evaluates the efficacy of the proposed loss (labelled as 'X') through SE(3) estimation task from noisy sets of 3D correspondences. It also shows comparisons with some varied robust losses which have been utilized in perception literature. $Cauchy$ (also known as Lorentzian) and GM (Geman-Mclure) are smooth, nonconvex losses that have been predominant in perception literature involving robust estimation. $Clip{-}L1$, GR (Geman-Reynolds) and $CauchyL1$ are nonsmooth, nonconvex losses — these have been utilized much less, as they are difficult to optimize in the general case. Instances in literature utilizing them have made specific assumptions on nature of residues or the loss objective, like linearity, convexity or similarly strong regularity (for instance [167, 161, 160, 142], *Section* 5.6) [20]. Since the task is one of SE(3) estimation, RANSAC based robust least squares 3D transform estimates have been evaluated as well ($SAC{-}L2A$ and $SAC{-}L2B$, with normalized thresholds of .025 and .0375 metres respectively to identify inliers) [21].

We utilzed datasets from [102, 103] which comprise of spatio-temporally contiguous depth

[20] Instances like [190] present application specific solutions. [190], which utilizes $Clip{-}L1$ loss, estimates SE(2) from 2D correspondences by exploring the 1D space of rotations. The approach will not generalize to arbitrary nonlinear residues and multivariate params.

[21] RANSAC is a discreet scheme, for robust fitting through random sampling. It is more useful for estimating models with low number of parameters.

images from varied indoor settings, and come with accurate pose ground truths. For ascertaining putative 3D correspondences between pairs of frames, SHOT descriptors were utilized. Keypoints were ascertained by uniform volumetric sampling of the pointclouds, and descriptors were matched based on proximity in the $L2$ sense. As consecutive frames in the datasets only had small motion between them, the correpondence sets were evaluated between pairs that were 10 and 25 frames apart (marked as '$step10$' and '$step25$' respectively). Thus the sets of putative correspondences had a signifcant number of outliers (false matches), as performance of local 3D descriptors like SHOT deteriorates sharply with increasing viewpoint changes. It also follows that correspondence sets arising from '$step25$' datasets had significantly lower inlier ratios than '$step10$' datasets.

SE(3) motion estimates between a pair of frames were then ascertained through the set of putative correspondences between them. The following objective was optimized :

$$\arg\min_{\theta} \sum_{\forall\{p_i,q_i\}} \rho\left(T(\theta)\, p_i - q_i\right) \tag{5.40a}$$

$$\boldsymbol{r}_i\left(\theta\right) = T(\theta)\, p_i - q_i \tag{5.40b}$$

(5.40) is the M-estimation objective (5.8) analyzed in *Section* 5.4, with residue specified as (5.40b). Above, $\{p_i, q_i\}$ indicates the set of putative correspondences between a pair of frames, and $T(\theta)$ is the SE(3) motion between them. ρ above, would be one of the losses indicated in *Figure* 5.5.

All the nonsmooth losses ($L1$, $Clip{-}L1$, GR, $CauchyL1$, X) were optimized using the method presented in *Section* 5.4 — as indicated earlier, these would have been difficult to optimize otherwise, due to the nonsmoothness and the residue function, $\boldsymbol{r}_i$, being nonlinear and nonconvex. The smooth, $L2$ based robust losses ($Cauchy$ and GM) were optimized using a method similar to [191, 147, 192]. The parameterizations were through Lie algebra. Graduated nonconvexity was employed in all optimizations. For experiments shown in *Figures* 5.5 and 5.6, the lowest (normalized) scale was kept at $\mathfrak{o} = .025$ metres (*Figure* 5.4 elucidates how regulating convexity through $\mathfrak{o}$ impacts performance).

Figure 5.5 shows the translation (top) and rotation (middle) errors, along with estimation success rates (bottom). An estimate was deemed successful if the rotation and translation errors were within thresholds of $.20$ metres and 20 degrees respectively. Better performance is indicated by lower values in the error plots and higher success rates in the bottom plot. The proposed loss, X, and $Clip{-}L1$ performed significantly better than the rest, across the board — both in terms of accuracy and robustness. Note that the improvements over the rest were larger when the inlier ratios were lower ('$step25$' datasets).

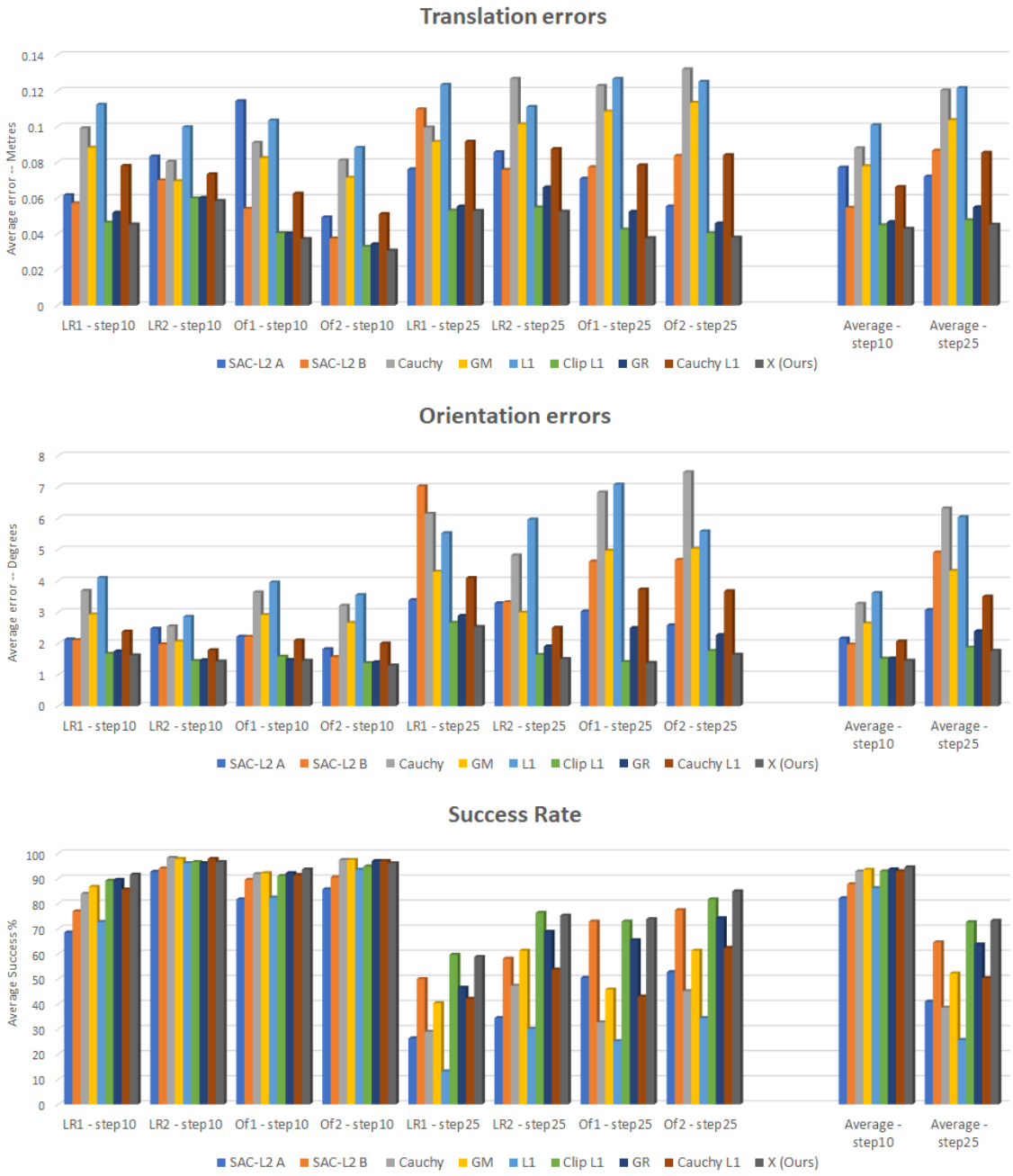


Figure 5.5: **Quantitative evaluations and comparisons :** Performance evaluation and comparison on SE(3) estimation task, from noisy sets of 3D correspondences. Average translation (top) and rotation (middle) errors in the SE(3) estimates are indicated, together with estimation success rates at the bottom. Datasets marked as '$step25$' have a significantly lower inlier ratio. The proposed loss, 'X', is compared with some varied robust losses in perception literature. $SAC{-}L2\,A$ and $SAC{-}L2\,B$ indicate RANSAC based discreet fitting with different thresholds. X and $Clip{-}L1$ performed significantly better than the rest, across the board. Also, losses which are both nonconvex and nonsmooth performed significantly better than the rest. X, $L1$, $Clip{-}L1$, GR and $CauchyL1$ were optimized using method presented in *Section* 5.4 — these would have been difficult to optimize otherwise, since they are nonsmooth and the residues involved are nonlinear and nonconvex.

Figure 5.6 shows the number of exact or near exact fitting data points / correspondences with respect to various estimates. For each loss, this was ascertained by using its transform estimate to evaluate the residues (5.40b), and counting the number of residues which were within a certain threshold (.025 metres). The results in *Figure* 5.6 corroborate the nonsmoothness property discussed in *Section* 5.2. X, $L1$, $Clip{-}L1$, GR and $CauchyL1$ all satisfy this property, and have significantly higher fit counts than the smooth robust losses, $Cauchy$ and GM. Along expected lines, X, together with $Clip{-}L1$, clearly has the highest fit count [22].

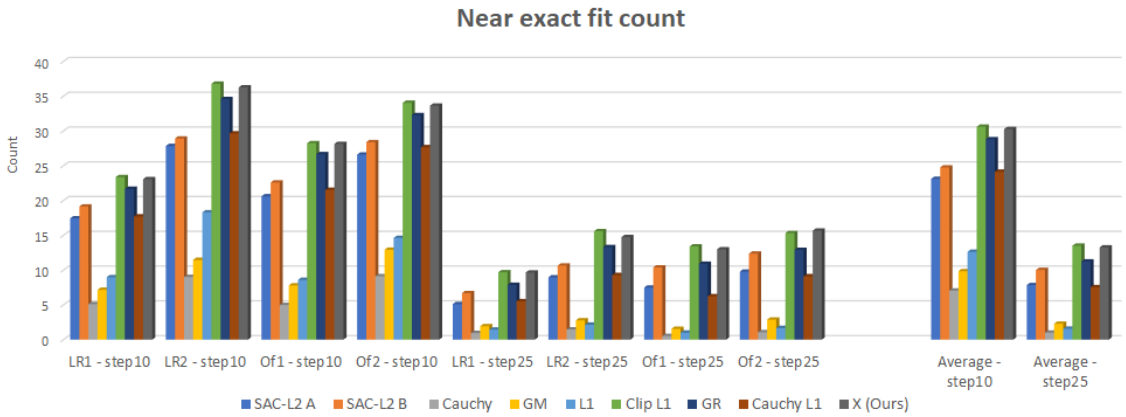


Figure 5.6: **Data fitting :** For various robust losses, we indicate the number of data points (correspondences) which fitted exactly or near-exactly to their model estimate (SE(3)). Results with RANSAC ($SAC{-}L2\,A$ and $SAC{-}L2\,B$, different thresholds) have been shown as well for perspective. Datasets marked as '$step25$' have a significantly lower inlier ratio. The proposed loss, X and $Clip{-}L1$ have clearly higher fit counts. In general, the nonsmooth robust losses had significantly higher fit counts than the smooth robust ones.

Note that the nonsmooth, nonconvex losses performed significantly better than the rest ($L1$ is nonsmooth but convex). The properties discussed in *Section* 5.2 support the results, given that the data had outliers and is likely distributed irregularly. The accurate, as well as robust, estimates were from losses X and $Clip{-}L1$ — both of which allow zero and near zero residues to have maximum impact, and reject larger ones. The two losses have notable differences though. Following *Section* 5.2 and *Figures* 5.1 and 5.2, $Clip{-}L1$ either allows data point residues to have maximum / equal impact on the estimate, or rejects them completely. It thus basically evaluates the median result post truncation, not allowing any tail influences at all. In contrast, X allows a smooth gradation for residues not close to zero – the weaker inliers are significantly downweighted, and the larger ones are effectively rejected. Thus while $Clip{-}L1$ can be slightly more robust, X can typically be expected to have a higher statistical efficiency. While the two performed similarly on the considered data, we do not

[22] The relatively low fit count for $L1$, especially in '$step25$' datasets, we suspect, is due to it's convexity and the presence of significant number of outliers. As the outliers go unsuppressed, they skew the estimate greatly from the truth (error plots in *Figure* 5.5). The RANSAC schemes have appreciable fit counts by virtue of their discrete fitting methodology.

expect this to be the case in general.

5.9 Conclusion

A loss function with properties well suited for robust, exact estimation was proposed. It performed promisingly in our experiments and compared well with popular robust losses in perception literature. To optimize M-estimation and structured estimation objectives with the proposed loss function, a robust optimization methodology with convergence assurances was proposed. The methodology is quite general and directly applies to an important class of nonsmooth, nonconvex losses and resulting objectives, that have been difficult to optimize. It supports block wise optimization for increased scalability and efficiency. A nonlinear least absolute deviations solver was developed as part of the proposed framework. The solver utilizes efficient and stable proximal operations. It is useful by itself as it can address general least absolute deviation based problems in a nonlinear setting. Future work would focus on utilizing the proposed loss and optimization methodology in various applications that require robustness — such as structured estimation (reconstruction for instance) and parameter / model learning (regression for instance).

CHAPTER 6

CONCLUDING COMMENTS

This dissertation discussed robust solutions to some fundamental 3D association problems, and more generally applicable methods for robust analysis. The presented approaches and methods were effective — they performed robustly on realistic data (particularly from challenging 3D modality), and compared well with the state of the art.

Particularly, it was well established that — *By leveraging 3D geometry at macro scales, it is possible to perform purely geometric analysis of real world 3D data that is robust in the face of noise, viewpoint changes, occlusions and partially overlapping content.*

Robustness to data challenges can have benefits that go beyond good performance gains. Doing away with restrictive assumptions and problem simplifications, not only broadens applicability and scope but can also compel novel use cases. In *Chapters* 3 and 4, we saw how this enabled operation in challenging, even unaddressed, settings and scenarios. And in *Chapters* 2 and 5, we saw how this led to design of more powerful tools for analysis.

In *Chapters* 3 and 4, we saw how utilization of macro scale 3D geometry enabled a purely geometric approach to perform on par with state-of-the-art methods based on RGB and RGB-D, besides outperforming 3D only baselines. It led to representations that had a high degree of robustness to noise, local ambiguities, sharp viewpoint changes, occlusions, partially overlapping content and related challenges. The approach afforded means to effectively capture 3D structural content — invariantly and robustly. It also admitted a geometric feature space that generalized well across varied environments.

The approach thus holds further potential; in recognition, detection and reconstruction tasks in particular. We look forward to utilize such features — which capture 3D context at macro scales, are robust and viewpoint invariant — in a deep neural network based framework for recognition and detection. It would also be interesting to utilize the methods in *Chapters* 3 and 4 for large scale freeform reconstructions, possibly from a repository of point clouds and range images — without requiring them to be spatio - temporally ordered or proximal. An open problem is to characterize and quantify *geometric ambiguity* in 3D point sets. This would prove useful in tackling structurally ambiguous scenes, such as ones pertaining to just parallel walls, or stairs.

In *Chapter* 5, we saw how robustness can be achieved at a more fundamental level, by addressing the optimization involved in estimation of statistics of interest. We studied a

novel robust loss. Through theoretical analysis and empirical validation, we ascertained how its particular kind of nonconvexity and nonsmoothness led to aggressive outlier suppression and more exact estimation. Its distinctive combination of properties seemed well suited for applications requiring robust parameter estimation. In conjunction, we also devised an effective optimization methodology for related, and important, class of nonsmooth, nonconvex losses and resulting objectives — these would have been difficult to optimize otherwise. As part of it, we developed a nonlinear least absolute deviations solver that would prove useful by itself, as least absolute deviations based approaches outperform their least squares and related counterparts in a variety of scenarios.

This opens up some very interesting possibilities. It enables us to readdress problems that require robustness with potentially better solutions. For instance, structured estimation problems such as ones that arise in 3D reconstruction and SLAM. Reconstruction methodologies in literature have predominantly utilized L_2 loss, which leads to a nonlinear least squares objective. The few robust methodologies in literature, have again utilized smooth (albeit nonconvex) losses. As observed in *Chapter* 5, nonsmooth nonconvex losses, and the proposed loss in particular, seem better suited for such problems. Understandably, we are looking forward to the prospects.

REFERENCES

[1] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3d local feature descriptors," *International Journal of Computer Vision*, pp. 1–24, 2015.

[2] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, 2010, pp. 998–1005.

[3] H. Bulow and A. Birk, "Spectral 6dof registration of noisy 3d range data with partial overlap," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2013.

[4] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga, "Fast registration based on noisy planes with unknown correspondences for 3-D mapping," *Robotics, IEEE Transactions on*, 2010.

[5] E. Olson, M. Walter, S. J. Teller, and J. J. Leonard, "Single-cluster spectral graph partitioning for robotics applications.," in *RSS*, 2005.

[6] A. Censi and S. Carpin, "Hsm3d: Feature-less global 6dof scan-matching in the hough/radon domain," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, IEEE, 2009, pp. 3899–3906.

[7] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.

[8] A. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3d point clouds," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, IEEE, vol. 1, 2006, pp. 1297–1304.

[9] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[10] B.-s. Kim, P. Kohli, and S. Savarese, "3d Scene Understanding by Voxel-CRF," in *2013 IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1425–1432.

[11] M. Labbe and F. Michaud, "Online global loop closure detection for large-scale multi-session graph-based slam," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, IEEE, 2014, pp. 2661–2666.

[12] E. Fernandez-Moral, W. Mayol-Cuevas, V. Arevalo, and J. González-Jiménez, "Fast place recognition with plane-based maps," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, IEEE, 2013, pp. 2719–2724.

[13] R. Cupec, E. K. Nyarko, D. Filko, A. Kitanov, and I. Petrovic, "Place recognition based on matching of planar surfaces and line segments," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 674–704, 2015.

[14] E. Olson, "Recognizing places using spectrally clustered local matches," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1157–1172, 2009.

[15] B. Steder, M. Ruhnke, S. Grzonka, and W. Burgard, "Place recognition in 3d scans using a combination of bag of words and point feature based relative pose estimation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, 2011, pp. 1249–1255.

[16] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[17] G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, IEEE, 1998, pp. 214–219.

[18] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[19] D. Erdogmus, U. Ozertem, and T. Lan, "Information theoretic feature selection and projection," in *Speech, Audio, Image and Biomedical Signal Processing using Neural Networks*, Springer, 2008, pp. 1–22.

[20] P. Kohli, P. H. Torr, *et al.*, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, 2009.

[21] L. Yang, P. Meer, and D. J. Foran, "Multiple class segmentation using a unified framework over mean-shift patches," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, IEEE, 2007, pp. 1–8.

[22] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, IEEE, 2007, pp. 1–8.

[23] U. Ozertem, D. Erdogmus, and R. Jenssen, "Mean shift spectral clustering," *Pattern Recognition*, vol. 41, no. 6, pp. 1924–1938, 2008.

[24] T. H. Kim, K. M. Lee, and S. U. Lee, "Learning full pairwise affinities for spectral segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 2101–2108.

[25] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 929–944, 2007.

[26] M. Surkala, K. Mozdren, R. Fusek, and E. Sojka, "Hierarchical blurring mean-shift," in *Advances Concepts for Intelligent Vision Systems*, Springer, 2011, pp. 228–238.

[27] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Computer Vision, International Conference on*, IEEE, vol. 1, 2001, pp. 438–445.

[28] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: A texture classification example," in *Computer Vision, International Conference on*, IEEE, 2003, pp. 456–463.

[29] J. Chacon and T. Duong, "Data-driven density derivative estimation, with applications to nonparametric clustering and bump hunting," *Electronic Journal of Statistics*, vol. 7, pp. 499–532, 2013.

[30] I. Horova, J. Kolacek, and K. Vopatova, "Full bandwidth matrix selectors for gradient kernel density estimate," *Computational Statistics & Data Analysis*, vol. 57, no. 1, pp. 364–376, 2013.

[31] D. Comaniciu, "An algorithm for data-driven bandwidth selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 2, pp. 281–288, 2003.

[32] R. Sawhney, H. Christensen, and G. Bradski, "Anisotropic agglomerative adaptive mean-shift," in *Proceedings of the British Machine Vision Conference*, BMVA Press, 2014.

[33] M. Á. Carreira-Perpiñán, "Gaussian Mean-Shift is an EM algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 5, pp. 767–776, 2007.

[34] K. Zhang, J. T. Kwok, and M. Tang, "Accelerated convergence using dynamic mean shift," in *Computer Vision–ECCV 2006*, Springer, 2006, pp. 257–268.

[35] S. Paris and F. Durand, "A topological approach to hierarchical segmentation using mean shift," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, IEEE, 2007, pp. 1–8.

[36] X.-T. Yuan, B.-G. Hu, and R. He, "Agglomerative mean-shift clustering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 2, pp. 209–219, 2012.

[37] A. Mayer and H. Greenspan, "An adaptive mean-shift framework for mri brain segmentation," *Medical Imaging, IEEE Transactions on*, vol. 28, no. 8, pp. 1238–1250, 2009.

[38] R. Jimenez-Alaniz, M. Pohi-Alfaro, V. Medina-Bafluelos, and O. Yaflez-Suarez, "Segmenting brain mri using adaptive mean shift," in *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, 2006, pp. 3114–3117.

[39] M.-H. Jeong, B.-J. You, Y. Oh, S.-R. Oh, and S.-H. Han, "Adaptive mean-shift tracking with novel color model," in *Mechatronics and Automation, 2005 IEEE International Conference*, 2005, pp. 1329–1333.

[40] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564–577, 2003.

[41] B. Leibe and B. Schiele, "Scale-invariant object categorization using a scale-adaptive mean-shift search," in *Pattern Recognition*, Springer, 2004, pp. 145–153.

[42] T. Vojir, J. Noskova, and J. Matas, "Robust scale-adaptive mean-shift for tracking," in *Image Analysis*, Springer, 2013, pp. 652–663.

[43] M. Carreira-Perpinan, "Fast nonparametric clustering with gaussian blurring mean-shift," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, New York, NY, USA: ACM, 2006, pp. 153–160.

[44] M. Surkala, K. Mozdren, R. Fusek, and E. Sojka, "Hierarchical evolving mean-shift," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 1593–1596.

[45] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Computer Vision–ECCV 2008*, Springer, 2008, pp. 705–718.

[46] J. Wang, B. Thiesson, Y. Xu, and M. Cohen, "Image and video segmentation by anisotropic kernel mean shift," in *Computer Vision-ECCV 2004*, Springer, 2004, pp. 238–249.

[47] T. Duong, A. Cowling, I. Koch, and M. Wand, "Feature significance for multivariate kernel density estimation," *Computational Statistics & Data Analysis*, vol. 52, no. 9, pp. 4225–4242, 2008.

[48] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *Sankhya - The Indian Journal of Statistics*, pp. 401–406, 1946.

[49] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring

ecological statistics," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, IEEE, vol. 2, 2001, pp. 416–423.

[50] C. M. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, IEEE, vol. 4, 2002, pp. 150–155.

[51] A. Asuncion and D. Newman, *UCI machine learning repository - http://archive.ics.uci.edu/ml/*, 2007.

[52] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Robotics and Automation*, 2001.

[53] F. Dellaert, "Monte-carlo em for data-association and its applications in computer vision," PhD thesis, Carnegie Mellon University, 2001.

[54] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: Science and Systems (RSS)*, 2009.

[55] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," in *International Symposium on Experimental Robotics*, 2010.

[56] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SfM and object labels," in *Computer Vision (ICCV), International Conference on*, 2013.

[57] T. Whelan, M. Kaess, J. Leonard, and J. McDonald, "Deformation-based loop closure for large scale dense RGB-D SLAM," in *IROS*, 2013.

[58] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching," in *Image Processing (ICIP), International Conference on*, 2011.

[59] A. Kowdle, S. N. Sinha, and R. Szeliski, "Multiple view object cosegmentation using appearance and stereo cues," in *ECCV*, 2012.

[60] D. Lin, S. Fidler, and R. Urtasun, "Holistic scene understanding for 3D object detection with rgbd cameras," in *ICCV*, 2013.

[61] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of the ACM*, 1964.

[62] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, 2001.

[63] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Intelligent Robots and Systems (IROS)*, 2012.

[64] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *ECCV*, 2010.

[65] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms for 3D registration," in *Robotics and Automation (ICRA)*, 2009.

[66] J. Folkesson, P. Jensfelt, and H. I. Christensen, "The M-space feature representation for SLAM," *Robotics, IEEE Transactions on*, 2007.

[67] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *Robotics and Automation*, 2013.

[68] M. Dou, L. Guan, J.-M. Frahm, and H. Fuchs, "Exploring high-level plane primitives for indoor 3d reconstruction with a hand-held RGB-D camera," in *Computer Vision - ACCV Workshops*, 2013.

[69] A. J. Trevor, J. Rogers, and H. I. Christensen, "Planar surface SLAM with 3D and 2D sensors," in *Robotics and Automation (ICRA)*, 2012.

[70] T. D. Stoyanov, M. Magnusson, H. Andreasson, and A. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations," *The International Journal of Robotics Research (IJRR)*, 2012.

[71] A. J. P. Bustos, T.-J. Chin, and D. Suter, "Fast rotation search with stereographic projections for 3D registration," in *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[72] H. Li and R. Hartley, "The 3D-3D registration problem revisited," in *Computer Vision (ICCV)*, IEEE, 2007.

[73] C. Olsson, F. Kahl, and M. Oskarsson, "The registration problem revisited: Optimal solutions from points, lines and planes," in *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[74] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Symposium on Geometry Processing*, 2005.

[75] J. Yang, H. Li, and Y. Jia, "Go-ICP : Solving 3D registration efficiently and globally optimally," in *Computer Vision (ICCV)*, 2013.

[76] E. Herbst, X. Ren, and D. Fox, "RGB-D flow: Dense 3D motion estimation using color and depth," in *ICRA*, IEEE, 2013.

[77] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Robotics and Automation (ICRA)*, IEEE, 2013.

[78] P. Henry, D. Fox, A. Bhowmik, and R. Mongia, "Patch volumes: Segmentation-based consistent mapping with RGB-D cameras," in *3D Vision (3DV), International Conference on*, 2013.

[79] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[80] C. Cagniart, E. Boyer, and S. Ilic, "Probabilistic deformable surface tracking from multiple videos," in *Computer Vision - ECCV*, 2010.

[81] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *ISRR*, 2011.

[82] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Visually bootstrapped generalized ICP," in *ICRA*, 2011.

[83] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision (IJCV)*, 2004.

[84] X. Ren, L. Bo, and D. Fox, "RGB-D scene labeling: Features and algorithms," in *Computer Vision Pattern Recognition (CVPR)*, 2012.

[85] B. Micusik and J. Kosecka, "Multi-view superpixel stereo in urban environments," *International Journal of Computer Vision*, 2010.

[86] M. Bleyer, C. Rother, and P. Kohli, "Surface stereo with soft segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[87] S. N. Sinha, D. Scharstein, and R. Szeliski, "Efficient high-resolution stereo matching using local plane sweeps," in *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[88] A. Bodis-Szomoru, H. Riemenschneider, and L. V. Gool, "Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels," in *Computer Vision and Pattern Recognition*, 2014.

[89] D. Gallup, J. M. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[90] Y. Eshet, S. Korman, E. Ofek, and S. Avidan, "DCSH - matching patches in RGBD images," in *Computer Vision (ICCV)*, 2013.

[91] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *ECCV*, 2010.

[92] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," in *ACM Transactions on Graphics (TOG)*, ACM, 2011.

[93] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Autonomous Robots*, 1998.

[94] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *International journal of pattern recognition and artificial intelligence*, 2004.

[95] L. Torresani, V. Kolmogorov, and C. Rother, "A dual decomposition approach to feature correspondence," *Pattern Analysis and Machine Intelligence (PAMI)*, 2013.

[96] T. Bailey, E. M. Nebot, J. Rosenblatt, and H. F. Durrant-Whyte, "Data association for mobile robot navigation: A graph theoretic approach," in *Robotics and Automation (ICRA)*, 2000.

[97] T. Caelli and T. Caetano, "Graphical models for graph matching: Approximate models and optimal algorithms," *Pattern Recognition Letters*, 2005.

[98] M. Cho and K. M. Lee, "Progressive graph matching: Making a move of graphs via probabilistic voting," in *CVPR*, 2012.

[99] C. Wang, L. Wang, and L. Liu, "Improving graph matching via density maximization," in *Computer Vision (ICCV)*, 2013.

[100] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation - supervoxels for point clouds," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2027–2034.

[101] D. Weikersdorfer, D. Gossow, and M. Beetz, "Depth-adaptive superpixels," in *International Conference on Pattern Recognition (ICPR)*, 2012, pp. 2087–2090.

[102] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *Robotics and automation (ICRA), 2014 IEEE international conference on*, IEEE, 2014, pp. 1524–1531.

[103] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015, pp. 5556–5565.

[104] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE robotics & automation magazine*, 2006.

[105] M. Bosse and R. Zlot, "Place recognition using keypoint voting in large 3d lidar datasets," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, IEEE, 2013, pp. 2677–2684.

[106] M. Magnusson, H. Andreasson, A. Nüchter, and A. J. Lilienthal, "Automatic appearance-based loop detection from three-dimensional laser data using the normal distributions transform," *Journal of Field Robotics*, vol. 26, no. 11-12, pp. 892–914, 2009.

[107] M. Himstedt and E. Maehle, "Geometry matters: Place recognition in 2D range scans using geometrical surface relations," in *Mobile Robots (ECMR)*, IEEE, 2015, pp. 1–6.

[108] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis, "A generative model for the joint registration of multiple point sets," in *European Conference on Computer Vision*, Springer, 2014, pp. 109–122.

[109] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[110] R. Paul and P. Newman, "Fab-map 3d: Topological mapping with spatial and visual appearance," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, 2010, pp. 2649–2656.

[111] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[112] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, 2015.

[113] S. Li and A. Calway, "Rgbd relocalisation using pairwise geometry and concise key point sets," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 6374–6379.

[114] S. Satkin and M. Hebert, "3dnn: Viewpoint invariant 3d geometry matching for scene understanding," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1873–1880.

[115] J. Valentin, M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. H. Torr, "Exploiting uncertainty in regression forests for accurate camera relocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4400–4408.

[116] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "DSAC - differentiable RANSAC for camera localization," *CoRR*, vol. abs/1611.05705, 2016.

[117] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation," in *Computer Vision (ICCV)*, 2017.

[118] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[119] K. Granström, T. B. Schön, J. I. Nieto, and F. T. Ramos, "Learning to close loops from range data," *The international journal of robotics research*, vol. 30, no. 14, pp. 1728–1754, 2011.

[120] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3d object recognition in cluttered scenes with local surface features: A survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2014.

[121] R. Sawhney, F. Li, and H. Christensen, "GASP: Geometric association with surface patches," in *3D Vision (3DV), International Conference on*, vol. 1, 2014, pp. 107–114.

[122] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, Q. Chen, N. K. Chowdhury, B. Fang, *et al.*, "A comparison of 3d shape retrieval methods based on a large-scale benchmark supporting multimodal queries," *Computer Vision and Image Understanding*, vol. 131, pp. 1–27, 2015.

[123] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 5, pp. 433–449, 1999.

[124] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, 2011, pp. 821–826.

[125] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5105–5114.

[126] Z. Koldovsky, P. Tichavsky, and E. Oja, "Efficient variant of algorithm fastica for independent component analysis attaining the cramér-rao lower bound," *IEEE Transactions on neural networks*, vol. 17, no. 5, pp. 1265–1277, 2006.

[127] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," *Advances in neural information processing systems*, pp. 487–493, 1999.

[128] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Computer Vision–ECCV 2010*, Springer, 2010, pp. 143–156.

[129] A. Kulesza and B. Taskar, "Determinantal point processes for machine learning," *arXiv preprint arXiv:1207.6083*, 2012.

[130] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time rgb-d camera relocalization," in *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, IEEE, 2013, pp. 173–179.

[131] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2930–2937.

[132] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image," 2016.

[133] A. Guzman-Rivera, P. Kohli, B. Glocker, J. Shotton, T. Sharp, A. Fitzgibbon, and S. Izadi, "Multi-output learning for camera relocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1114–1121.

[134] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," *arXiv preprint arXiv:1702.04405*, 2017.

[135] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Scenenet: Understanding real world indoor scenes with synthetic data," *arXiv preprint arXiv:1511.07041*, 2015.

[136] J. P. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. H. Torr, "Mesh based semantic modelling for indoor and outdoor scenes," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, IEEE, 2013, pp. 2067–2074.

[137] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," *CoRR*, vol. abs/1703.07971, 2017.

[138] Y. Sharon, J. Wright, and Y. Ma, "Minimum sum of distances estimator: Robustness and stability," in *American Control Conference, 2009. ACC'09.*, IEEE, 2009, pp. 524–530.

[139] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 764–766, 2013.

[140] F. R. Hampel, "The influence curve and its role in robust estimation," *Journal of the american statistical association*, vol. 69, no. 346, pp. 383–393, 1974.

[141] T. Kanamori, S. Fujiwara, and A. Takeda, "Breakdown point of robust support vector machine," *arXiv preprint arXiv:1409.0934*, 2014.

[142] Y.-l. Yu, Ö. Aslan, and D. Schuurmans, "A polynomial-time form of robust regression," in *Advances in Neural Information Processing Systems*, 2012, pp. 2483–2491.

[143] P. J. Huber, "Robust estimation of a location parameter," *The annals of mathematical statistics*, pp. 73–101, 1964.

[144] M. Nikolova, "Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers," 3, vol. 40, SIAM, 2002, pp. 965–994.

[145] ——, "Either fit to data entries or locally to prior: The minimizers of objectives with nonsmooth nonconvex data fidelity and regularization," in *International Conference on Scale Space and Variational Methods in Computer Vision*, Springer, 2011, pp. 110–121.

[146] ——, "Energy minimization methods," *Handbook of Mathematical Methods in Imaging*, pp. 157–204, 2015.

[147] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.

[148] J. Idier, "Convex half-quadratic criteria and interacting auxiliary variables for image restoration," *IEEE transactions on image processing*, vol. 10, no. 7, pp. 1001–1009, 2001.

[149] M. Allain, J. Idier, and Y. Goussard, "On global and local convergence of half-quadratic algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1130–1142, 2006.

[150] M. Nikolova and R. H. Chan, "The equivalence of half-quadratic minimization and the gradient linearization iteration," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1623–1627, 2007.

[151] R. He, W.-S. Zheng, T. Tan, and Z. Sun, "Half-quadratic-based iterative minimization for robust sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 2, pp. 261–275, 2014.

[152] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[153] H.-J. M. Shi, S. Tu, Y. Xu, and W. Yin, "A primer on coordinate descent algorithms," *arXiv preprint arXiv:1610.00040*, 2016.

[154] M. J. Powell, "On search directions for minimization algorithms," *Mathematical Programming*, vol. 4, no. 1, pp. 193–201, 1973.

[155] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 57–77, 2016.

[156] C. Xu, Z. Lin, Z. Zhao, and H. Zha, "Relaxed majorization-minimization for non-smooth and non-convex optimization.," in *AAAI*, 2016, pp. 812–818.

[157] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[158] ——, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *Journal of Scientific Computing*, vol. 72, no. 2, pp. 700–734, 2017.

[159] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, 2001.

[160] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[161] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier analysis and applications*, vol. 14, no. 5, pp. 877–905, 2008.

[162] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[163] P. Ochs, Y. Chen, T. Brox, and T. Pock, "Ipiano: Inertial proximal algorithm for nonconvex optimization," *SIAM Journal on Imaging Sciences*, vol. 7, no. 2, pp. 1388–1419, 2014.

[164] H. Li and Z. Lin, "Accelerated proximal gradient methods for nonconvex programming," in *Advances in neural information processing systems*, 2015, pp. 379–387.

[165] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[166] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific Journal of Optimization*,

[167] P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock, "On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision," *SIAM Journal on Imaging Sciences*, vol. 8, no. 1, pp. 331–372, 2015.

[168] Y. Yu, X. Zheng, M. Marchetti-Bowick, and E. Xing, "Minimizing nonconvex non-separable functions," in *Artificial Intelligence and Statistics*, 2015, pp. 1107–1115.

[169] X. Chen, "Smoothing methods for nonsmooth, nonconvex minimization," *Mathematical programming*, pp. 1–29, 2012.

[170] J. V. Burke, A. S. Lewis, and M. L. Overton, "A robust gradient sampling algorithm for nonsmooth, nonconvex optimization," *SIAM Journal on Optimization*, vol. 15, no. 3, pp. 751–779, 2005.

[171] A. S. Lewis and M. L. Overton, "Nonsmooth optimization via quasi-newton methods," *Mathematical Programming*, pp. 1–29,

[172] K. Lange, *MM Optimization Algorithms*. SIAM, 2016.

[173] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of computational and graphical statistics*, vol. 9, no. 1, pp. 1–20, 2000.

[174] P. Tseng, "An analysis of the em algorithm and entropy-like proximal point methods," *Mathematics of Operations Research*, vol. 29, no. 1, pp. 27–44, 2004.

[175] Y.-X. Yuan, "Recent advances in trust region algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 249–281, 2015.

[176] J. Mairal, "Incremental majorization-minimization optimization with application to large-scale machine learning," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 829–855, 2015.

[177] D. Davis and W. Yin, "Faster convergence rates of relaxed peaceman-rachford and admm under regularity assumptions," *Mathematics of Operations Research*, 2017.

[178] P. Giselsson and S. Boyd, "Diagonal scaling in douglas-rachford splitting and admm," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, IEEE, 2014, pp. 5033–5039.

[179] N. Parikh, S. Boyd, *et al.*, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[180] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, Springer, 2011, pp. 185–212.

[181] D. W. Peaceman and H. H. Rachford Jr, "The numerical solution of parabolic and elliptic differential equations," *Journal of the Society for industrial and Applied Mathematics*, vol. 3, no. 1, pp. 28–41, 1955.

[182] G. Li, T. Liu, and T. K. Pong, "Peaceman–rachford splitting for a class of nonconvex optimization problems," *Computational Optimization and Applications*, vol. 68, no. 2, pp. 407–436, 2017.

[183] B. He, H. Liu, Z. Wang, and X. Yuan, "A strictly contractive peaceman–rachford splitting method for convex programming," *SIAM Journal on Optimization*, vol. 24, no. 3, pp. 1011–1040, 2014.

[184] B. He, F. Ma, and X. Yuan, "Convergence study on the symmetric version of admm with larger step sizes," *SIAM Journal on Imaging Sciences*, vol. 9, no. 3, pp. 1467–1501, 2016.

[185] Y. Gu, B. Jiang, and D. Han, "A semi-proximal-based strictly contractive peaceman-rachford splitting method," *arXiv preprint arXiv:1506.02221*, 2015.

[186] L. Qi and J. Sun, "A trust region algorithm for minimization of locally lipschitzian functions," *Mathematical Programming*, vol. 66, no. 1, pp. 25–43, 1994.

[187] A. Blake and A. Zisserman, *Visual reconstruction*. 1987.

[188] E. Hazan, K. Y. Levy, and S. Shalev-Shwartz, "On graduated optimization for stochastic non-convex problems," in *International Conference on Machine Learning*, 2016, pp. 1833–1841.

[189] H. Mobahi and J. W. Fisher, "On the link between gaussian homotopy continuation and convex envelopes," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, 2015, pp. 43–56.

[190] E. Ask, O. Enqvist, L. Svärm, F. Kahl, and G. Lippolis, "Tractable and reliable registration of 2d point sets," in *European Conference on Computer Vision*, Springer, 2014, pp. 393–406.

[191] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM Journal on Scientific computing*, vol. 27, no. 3, pp. 937–966, 2005.

[192] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conference on Computer Vision*, Springer, 2016, pp. 766–782.